



Universidad
Carlos III de Madrid

PROYECTO FIN DE CARRERA

APLICACIÓN DE LAS NUEVAS
TECNOLOGÍAS WEB PARA LA
CLASIFICACIÓN DE CONTENIDOS EN
REDES SOCIALES

Autor: Pablo Rodríguez Fernández

Tutor: Pablo Acuña

Leganés, 26 de octubre de 2012

Título: *Aplicación de las nuevas tecnologías web para la clasificación de contenidos en redes sociales*

Autor: *Pablo Rodríguez Fernández*

Director: *Pablo Acuña*

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día 26 de Octubre de 2012 en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

Quiero agradecer todo el apoyo que he recibido en primer lugar a mi familia, por que ha sido el punto más importante durante estos años de carrera que han sido difíciles en muchos aspectos y que han estado apoyándome hasta el último día. A mis amigos, con los que he compartido grandes momentos en mi época como universitario y en parte ellos fueron la inspiración que me dio a entrar a la universidad y la posibilidad de conocer a grandes personas como han sido compañeros de clase estos casi cuatro años.

Una etapa que ha sido bastante dura y difícil por factores educativos, como los cambios de planes de estudios a las nuevas titulaciones de Bolonia y que no se nos olvidará nunca y también por el factor económico, sacarse una carrera con una crisis como la que hemos estado pasando. Aun así ha sido una bonita etapa y más aún el final de ver, como superar todo esto y poder llegar a la meta.

Ahora espero poder emprender un camino de posibilidades para conseguir lo que tanto deseamos, un trabajo equitativo al esfuerzo realizado durante estos años.

No quiero tampoco dejar sin dar agradecimientos a Pablo Acuña, mi tutor de este proyecto que ha sido un placer tenerle estos meses y con el cual me he sentido muy bien realizando el proyecto gracias a las aportaciones que me ha dado.

A todos en definitiva, gracias por esta gran etapa de mi vida, la de la Universidad.

Índice general

1. INTRODUCCIÓN Y OBJETIVOS	17
1.1 Introducción.....	17
1.2 Objetivos.....	20
1.3 Fases del desarrollo	20
1.4 Estructura de la memoria.....	21
2. ESTADO DEL ARTE.....	23
2.1 ¿Qué es “data curation”?	24
2.2 Redes sociales.....	25
2.2.1 Twitter	26
2.2.2 Twitter GET API.....	27
2.2.3 Twitter Streaming API.....	27
2.2.4 Twitter Search API.....	28
2.2.5 Diferencias entre Search API y Streaming API.....	30
2.2.6 ¿Qué API vamos a usar en nuestra aplicación?.....	31
2.3 Tecnologías	32
2.3.1 HTML.....	32
2.3.2 HTML5	34
2.3.3 CSS (Cascading Styles Sheet/Hojas de estilo en cascada).....	41
2.3.4 Javascript y JQuery.....	44
2.3.5 PHP (Hypertext Preprocessor)	44
2.3.6 Las tecnologías a usar.....	46
2.4 Aplicaciones existentes	47
2.4.1 Storify	47
2.4.2 Percolate.....	48
2.4.3 Focus.io	49
2.4.4 Summify	50
2.4.5 Comparativas.....	51
3. DESARROLLO DE LA APLICACIÓN.....	53
3.1 Introducción.....	53
3.2 Análisis y extracción de requisitos.....	54
3.3 Diseño e Implementación.....	55
3.3.1 Diseño de las funcionalidades de la aplicación	56

3.3.2. La interfaz gráfica de la página inicial.....	58
3.3.2 La interfaz y funcionamiento del almacenamiento.....	62
3.3.3 La implementación.....	62
3.3.3.1 El cuerpo de la aplicación.....	62
3.3.3.2 El buscador de tweets.....	63
3.3.3.3 Función Drag&Drop.....	64
3.3.3.4 El Almacenamiento en LocalStorage.....	65
3.3.3.5 Las historias guardadas y sus funciones.....	66
3.3.3.6 Funciones de las historias guardadas.....	67
3.3.4 Otras secciones de la aplicación.....	68
3.4 Fase de pruebas	69
3.4.1 Búsqueda de tweets.....	70
3.4.2 Edición de los usuarios.....	71
3.4.3 Guardado de datos	73
3.4.4 Compartición y/o visualización de los datos.....	73
4. PRESUPUESTO	77
4.1 Introducción.....	77
4.2 El presupuesto.....	79
5. CONCLUSIONES.....	81
5.1 Conclusiones del Proyecto.....	81
5.2 Conclusiones personales	82
5.3 Desarrollos y mejoras futuras	83

Índice de figuras

Figura 1 - Porcentaje de usuarios en las redes sociales en España.....	19
Figura 2 - Fases del desarrollo en cascada sociales en España	20
Figura 3 - La web de Twitter	26
Figura 4 - Ejemplo de código en HTML5 y sus nuevas etiquetas	33
Figura 5 - Cambios en la estructura de HTML con HTML5	41
Figura 6 - Ejemplo de código CSS.....	41
Figura 7 - Vista de la aplicación Storify.....	47
Figura 8 - Vista de la aplicación Percolate.....	48
Figura 9 - Vista de la aplicación Focus.io	49
Figura 10 - Vista de la aplicación Summify	50
Figura 11 - Funcionalidades de la aplicación web en HTML5	56
Figura 12 - Funcionalidades para la creación de una o más historias	57
Figura 13 - Esquema o boceto de nuestra aplicación web	58
Figura 14 - Cabecera de la aplicación desarrollada	58
Figura 15 - El motor de búsqueda	59
Figura 16 - Zona de curación de la aplicación	60
Figura 17 - La aplicación al completo	61
Figura 18 - Página principal de la aplicación	62
Figura 19 - Código HTML5 de la aplicación.....	62
Figura 20 - Código JQuery para la búsqueda de tweets en nuestra aplicación	63
Figura 21 - Código JQuery para el borrado de tweets.....	63
Figura 22 - Código en Javascript para permitir el Drag&Drop de los tweets	64
Figura 23 - El código Javascript del almacenamiento en local	65
Figura 24 - Vista de las historias guardadas en la aplicación.....	66
Figura 25 - Codigo de la obtención de datos de Local Storage	67
Figura 26 - Codigo del borrado de datos en Local Storage.....	68
Figura 27 - Vista de la sección “Acerca de” de la aplicación web	68
Figura 28 - Paso 1 – Búsqueda de tweets	70
Figura 29 - Paso 2 – Resultados de una palabra clave	70
Figura 30 - Paso 2 – Resultados de dos palabras claves	71
Figura 31 - Paso 3 – Arrastrado de tweets en la historia	71
Figura 32 - Paso 3 – Arrastrado de tweets con multihistorias	72

Figura 33 - Paso 4 – Guardado de datos	73
Figura 34 - Paso 5 – Visualización de las historias	74
Figura 35 - Paso 5 -Visualización de las historias y botones de borrado	75

Índice de tablas

Tabla 1 - Operadores de la función Search API.....	29
Tabla 2 - Código Json que nos devuelve Twitter en sus búsquedas.....	30
Tabla 3 - Cabecera de HTML5.....	33
Tabla 4 - Las etiquetas de HTML5 y sus cambios.....	39
Table 5 - Sintaxis de CSS.....	43
Tabla 6 - Comparativa de aplicaciones y nuestro proyecto.....	51

Resumen

Proyecto Fin de Carrera que presenta la aplicación de las nuevas tecnologías Web para la clasificación de contenidos en redes sociales, tratando dicha información mediante la técnica de “data curation”.

En el documento se describe a fondo de qué trata esta técnica. Principalmente, la aplicación Web permitirá crear entradas de historias con datos insertados por el propio usuario y procedentes de redes sociales.

Se ha desarrollado una aplicación Web que mediante un motor de búsqueda utilizando APIs nos permite obtener información con cierta palabra clave que se haya insertado, y así complimentar el usuario, la historia o noticia con los resultados que obtenemos en la búsqueda.

Para tener una referencia, se describen algunas de las aplicaciones ya existentes que utilizan esta técnica y su funcionamiento.

El desarrollo de dicha aplicación ha sido realizado con los lenguajes de programación Web, HTML5, Json, CSS3 con funciones JQuery y Javascript. Para su posterior guardado, se usa el almacenamiento local, donde podremos recuperarlo o visualizarlo cuando se desee.

En la fase de pruebas del proyecto se han creado historias o noticias añadiendo información, con ‘tweets’ de la red social Twitter, además de información escrita por el propio usuario como es el título de la historia o noticia.

También permite crear más de una a la vez, guardarlo, borrar todo o borrar únicamente aquella que se desee.

Finalmente, en el capítulo 5 se incluyen conclusiones del trabajo realizado, así como conclusiones personales y algunas sugerencias de mejoras futuras a la aplicación presentada.

Palabras Clave

Redes Sociales; Twitter; Aplicación Web; Curación de Datos.

Abstract

This project presents the development of a web application that allows a collaborative classification of content published in social networking sites through “data curation”.

Initially, I include background knowledge about data curation technique and its characteristics. Mainly, the web application allows creating stories with content selected by the user extracted from social networks.

The developed web application includes a search engine that connects to APIs provided by social networks, allowing the retrieval of information according to a keyword entered by the user. Based on this information, the user can select items and create stories inside the application.

I present the standards and technologies used for the development of the application, including HTML5, JSON, CSS3, Javascript and jQuery library, as well as the methods used for local storage. Moreover, I include a description of current web site applications that allow data curation, analyzing their strengths and weaknesses for better defining our application.

I present some example scenarios performed on the application for testing purposes, such as creation of stories with tweets retrieved from Twitter, including content added by the user (e.g. the title of the story). Finally, the user is able to create several stories, save them, delete a story or all stories created at once.

Finally, I include some conclusions about the work done, as well as some personal conclusions about the process and suggestions for improving the web application presented.

Keywords

Social Networks; Twitter; Web Application; Data Curation.

Capítulo 1

Introducción y objetivos

1.1 Introducción

Para tratar en primer lugar el proyecto a desarrollar, hay que hacer mención especial al origen de las redes sociales, como han evolucionado e influido tanto en España como en el resto del mundo, y posteriormente ver qué solución se le ha aplicado a la gran cantidad de información que está surgiendo hasta hoy.

Durante los últimos años, el mundo se ha visto influenciado por las conocidas “redes sociales”. Las redes sociales son aquellas plataformas que permiten conectar a las personas con sus amigos y familiares, incluso creándose de alguna manera nuevos vínculos o nuevos contactos, con el fin de compartir o conocer información relevante de especial interés a los usuarios, que pueden englobar contenido de blogs, wikis, foros, chats, y mensajería en un mismo entorno.

Estas redes de relaciones personales sociabilizan, apoyan e informan, dan sentido a una identidad. La mayoría de las veces se comunican por intereses comunes que suelen ser a través de proyectos, sintiendo una pertenencia a un grupo con una cultura común. Las redes sociales suelen tener una comunicación entre los usuarios de uno a todos y de uno a uno. El fin que conllevan es la realización de actividades para interactuar entre los integrantes que proporcionan una ayuda emotiva y cognitiva.

Las redes sociales han ganado terreno de tal forma que son un punto de inflexión muy importante para profesionales, artistas, marcas y personas físicas ya que pueden suponerle tanto ventajas a la hora de compartir información o iniciar nuevas relaciones personales o profesionales, como inconvenientes en la privacidad de información personal, o contenidos publicados que pueden dañar la imagen de una compañía en el ámbito empresarial, entre otros muchos.

Una de las mayores aportaciones que han producido las redes sociales es la gran cantidad de información que se genera a través de las distintas redes existentes. Desde vídeos, imágenes, noticias, textos, grupos, etc., resulta en una sobresaturación de información, creando la necesidad de analizarla, procesarla y categorizarla en diferentes ámbitos, ya sea personal como Facebook¹ o Tuenti², temático como Youtube³ para vídeos, Flickr⁴ para imágenes/fotografías, noticias en general como meneame.net⁵, o de ámbito profesional como LinkedIn⁶ que han comenzado a distinguirse de un gran abanico de las miles existentes.

Según ciertos estudios, España es uno de los países en el que se dedica más tiempo a las redes sociales, siendo indiferente la temática utilizada. En un [ranking elaborado por ONTSI](#) (Observatorio Nacional de las Telecomunicaciones y de la Sociedad de la Información), España se sitúa en el tercer puesto del ranking por detrás de los Estados Unidos y Reino Unido, con un 77% de usuarios activos en al menos alguna de las redes sociales existentes.

¹ www.facebook.com

² www.tuenti.com

³ www.youtube.com

⁴ www.flickr.com

⁵ www.meneame.net

⁶ www.linkedin.com

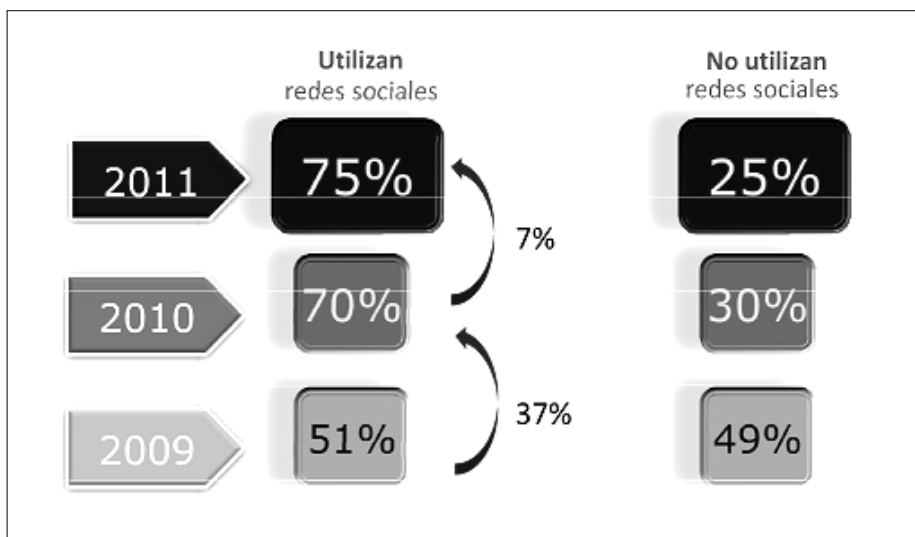


Figura 1 - Porcentaje de usuarios en las redes sociales en España

En los últimos años se ha visto el incremento de su uso, como muestra la Figura 1 del estudio realizado por la empresa IAB Spain⁷. En estos casos, donde existe una saturación de información en las redes sociales, se ha llevado a cabo la creación de otras redes para procesar dicha información. De esta forma, se crean comentarios, historias o noticias a raíz de lo que los usuarios compartan en otras redes como Twitter⁸, Facebook, Youtube, Blogger⁹, etc., para reducir dicha sobrecarga de información.

Por eso se ha desarrollado una aplicación web que realice la función de selección de información relevante que más nos interese.

Este trabajo de investigación está basado en un proyecto de fin de carrera previo que realizaba una funcionalidad similar, pero con ciertas deficiencias como: limitación en el número de historias creadas, permitiendo únicamente construir 2 historias en un momento indicado; almacenamiento local temporal de las historias, que no permitía al usuario almacenar historias completas de manera local o visualizar diferentes historias realizadas previamente. Este proyecto de fin de carrera toma como base dicho proyecto, pero en base a un análisis y desarrollo nuevos mejora su implementación y provee funcionalidades más completas para la curación de datos.

⁷ www.slideshare.net/IAB_Spain/iii-estudio-sobre-redes-sociales-en-internet

⁸ <http://www.twitter.com>

⁹ <http://www.blogger.com>

1.2 Objetivos

El tratamiento de la información y el contenido en las redes sociales se está comenzando a conocer mediante la técnica de “data curation”, traducido como “curación de datos” o “tratamiento de datos”.

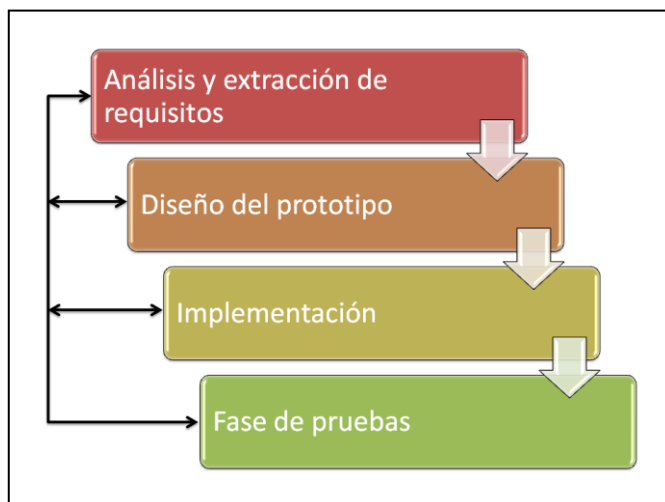
El objetivo principal del presente trabajo es la creación de una aplicación Web para permitir la curación de datos obtenidos de las redes sociales.

Los objetivos secundarios son los siguientes:

- Estudiar las técnicas de ‘data curation’ y el modo de aplicarlo en los datos obtenidos de las redes sociales.
- Analizar las herramientas proporcionadas por las redes sociales para obtener la información publicada por los usuarios de la misma.
- Desarrollar una aplicación Web que nos permita seleccionar en distintas categorías el contenido obtenido de las redes sociales, creando historias a través del contenido generado por usuarios de las redes sociales.

1.3 Fases del desarrollo

Las fases en las que consta este proyecto de fin de carrera son: Análisis y extracción de requisitos, Diseño del prototipo y evoluciones, Implementación y Fase de pruebas.



Con estas fases siguiendo el modelo en cascada, se lleva a cabo todo el desarrollo, de tal manera que si se necesita hacer alguna modificación, se permite volver a cualquiera de las fases anteriores.

Figura 2 - Fases del desarrollo en cascada sociales

CAPÍTULO 1

En el análisis y extracción de requisitos, se analiza la situación en el que se desarrolla el problema para proponer una solución en forma de Aplicación Web.

Mediante el diseño del prototipo, se recopilan las posibles funcionalidades para la aplicación Web, apariencia y la interfaz que se muestra al usuario.

La implementación se realiza mediante HTML, hojas de estilo CSS, y lenguajes para crear las funcionalidades interactivas, JQuery, etc.

1.4 Estructura de la memoria

A continuación se detallan todos los apartados en los que consta este Proyecto Fin de Carrera, describiendo los contenidos de la memoria.

El documento está dividido en 5 capítulos, donde se describen todos los aspectos necesarios para la implementación y elaboración de la aplicación Web.

Capítulo I: Contempla información no técnica, donde se puede encontrar una descripción breve del proyecto, los objetivos y la situación que ha impulsado a su creación.

Capítulo II: Incluye información del Estado del Arte, donde se define el término ‘data curation’ que es la clave principal del proyecto y el método por el que nos vamos a regir para su desarrollo. Además, se analizan las diferentes redes que lo usan. También la descomposición de la red social Twitter, que ha sido con la que se ha trabajado y que sirvió de base de la curación de datos para crear las historias o noticias.

Capítulo III: En el desarrollo de la aplicación, se puede ver un esbozo completo de como ha sido implementada toda la aplicación. Se describen todos los requisitos principales que debe cumplir la aplicación detallados completamente, así como las funcionalidades que tiene. También se comenta las partes en las que consta, explicando para que sirven así como parte del código fuente que tiene las funciones y como actúan. Y finalmente una batalla de pruebas con los pasos a seguir para usarla.

Capítulo IV: Se dedica exclusivamente al presupuesto de realización del proyecto para saber los costes tanto de personal, recursos y otros costes que se han tenido en cuenta para la puesta en funcionamiento de Charlify.

Capítulo V: En él se realiza una reflexión con las conclusiones que se han obtenido durante su desarrollo, tanto del proyecto en sí, como personales y de posibles mejoras para un futuro.

En otros apartados como el glosario y las referencias, se ha incluido información referente a siglas y acrónimos utilizados en la memoria, además de las referencias bibliográficas que se han citado en todo el documento.

Capítulo 2

Estado del Arte

En este apartado se hace alusión principalmente a la técnica principal en la que se centra este proyecto, el “data curation” o curación de datos, su definición, cómo funciona y dónde se puede aplicar.

También se hace un repaso a la red social a tratar para el desarrollo de este proyecto. En este caso, se trata de Twitter, a partir del cual se realiza el “data curation”.

Se describen algunas redes sociales y aplicaciones Web que tratan la curación de datos, como son Storify¹⁰, Percolate¹¹, Focus.io¹², Summify¹³, entre otros muchos existentes.

Repaso además a las diferentes tecnologías de desarrollo web, como HTML5, PHP, JSON, Javascript y CSS a tratar en este proyecto. Se citan y se comentan por qué se usan y para qué.

¹⁰ <http://storify.com/>

¹¹ <http://percolate.com/>

¹² <http://focus.io/>

¹³ <http://summify.com/>

Finalmente, una breve descripción de como se tratan los datos de las historias o publicaciones que se crean, la forma de guardar los datos, que en este caso se trata de un almacenamiento local.

2.1 ¿Qué es “data curation”?

Curation es el acto deliberado de organizar los contenidos en torno a un tema de manera que informa y educa a su público.

El concepto “data curation” se puede definir como la extracción de información importante de textos, tales como artículos de investigación o de cualquier otro tipo, para convertirlos a un formato electrónico.

En términos generales, “curation” denomina las actividades y procesos realizados para crear, administrar, mantener y validar un componente.

“La curación de datos, es la gestión activa y permanente de los datos de investigación a través de su ciclo de vida de interés y utilidad para el aprendizaje, la ciencia y la educación” – Carole Palmer, UIUC GSLIS

La curación o “curation” se basa en tres pasos:

- 1. Selección de contenido:** El administrador (o “curador”, como se definiría en un término más técnico) se encarga de identificar los contenidos y hacer una selección o clasificación. Reúne por tanto, el contenido en torno a un tema dado. Este punto es la clave principal del proceso de “curation”, que puede ser manual o automatizado.
- 2. Edición del contenido:** Se encarga de organizar, estructurar y dar formato al contenido. Aquí, al contenido seleccionado en la primera fase se le da un valor más real, contextualizando o/y unificándolo en torno a un tema. Estos desarrollos se realizan de acuerdo a los criterios de selección establecidos previamente.
- 3. Intercambio:** Se le conoce como “sharing” o compartir los datos. Los contenidos se ponen a disposición de todos los usuarios en la plataforma de curación, a través del curador o administrador, que lo difundirá para un público interesado con el tema a tratar.

CAPÍTULO 2

La curación permite un ahorro de tiempo al usuario durante la búsqueda de información en la red, evitando así la clasificación y selección de información que ya se ha realizado mediante este proceso, y donde los resultados se muestran en relación a lo que el usuario busca. Los contenidos están por tanto, organizados y completamente tematizados. También puede mostrarse contenidos a veces ignorados o poco accesibles desde los motores de búsqueda de Internet.

La curación de datos incluye actividades como la preservación digital, pero también aquellas que añaden valor a los datos para que puedan ser descubiertos o reutilizados. Es decir, **reutilizar información ya compartida o publicada en diferentes redes sociales, añadiéndole más información útil para los usuarios.**

Para tener una idea de curación de datos, podemos seleccionar contenidos Web (enlaces, vídeos, sonidos, imágenes) en una colección o conjunto coherente para editarlo. Por ejemplo, acerca de los acontecimientos en Egipto el año pasado, hubo una gran cantidad de tweets, estados de Facebook, fotos en Flickr y vídeos de YouTube publicados en todo el mundo a través de Internet. Pero para consolidarlo y crear contenido atractivo y con un valor añadido, debemos realizar "data curation" y construir una historia mediante la selección de la información que es más significativa.

2.2 Redes sociales

Las redes sociales son aquellas estructuras compuestas por grupos de personas, conectadas por uno o varios tipos de relaciones, con intereses en común o que comparten conocimientos.

En este apartado se analiza la red social que da origen a los datos externos de los que se nutre nuestra aplicación: Twitter. Esta red social está emergiendo cada día más, y nos ofrece numerosas posibilidades para los desarrolladores con diversas APIs disponibles para todos.

2.2.1 Twitter

Esta red social que funciona como una plataforma de micro-blogging, se creó en marzo de 2006, con sede en San Francisco aunque fue creada en California posteriormente como Twitter Inc. en 2007. Se estima que tiene unos 200 millones de usuarios, publicándose más de 65 millones de ‘tweets’ al día y unas 800.000 búsquedas diarias en la red.

Se centra básicamente en un contenido limitado por 140 caracteres, de los cuales pueden incluirse enlaces externos a otras páginas o adjuntar imágenes que se cargarán en los propios servidores de twitter, a través de plataformas como “twitpic”.

Los usuarios de la red social pueden realizar la función de seguimiento entre unos y otros, para ver sus publicaciones. También existe otra función llamada “retweets” donde los seguidores comparten la información de otros entre sus propios seguidores a través de su twitter.

Además, la red proporciona herramientas a los desarrolladores para crear aplicaciones utilizando sus datos. Estas herramientas se llaman APIs (Application Program Interface, en inglés), que proveen de funciones y métodos para interactuar con la información proporcionada por Twitter. Para ello pone a disposición del desarrollador tres APIs, que se comentan a continuación.

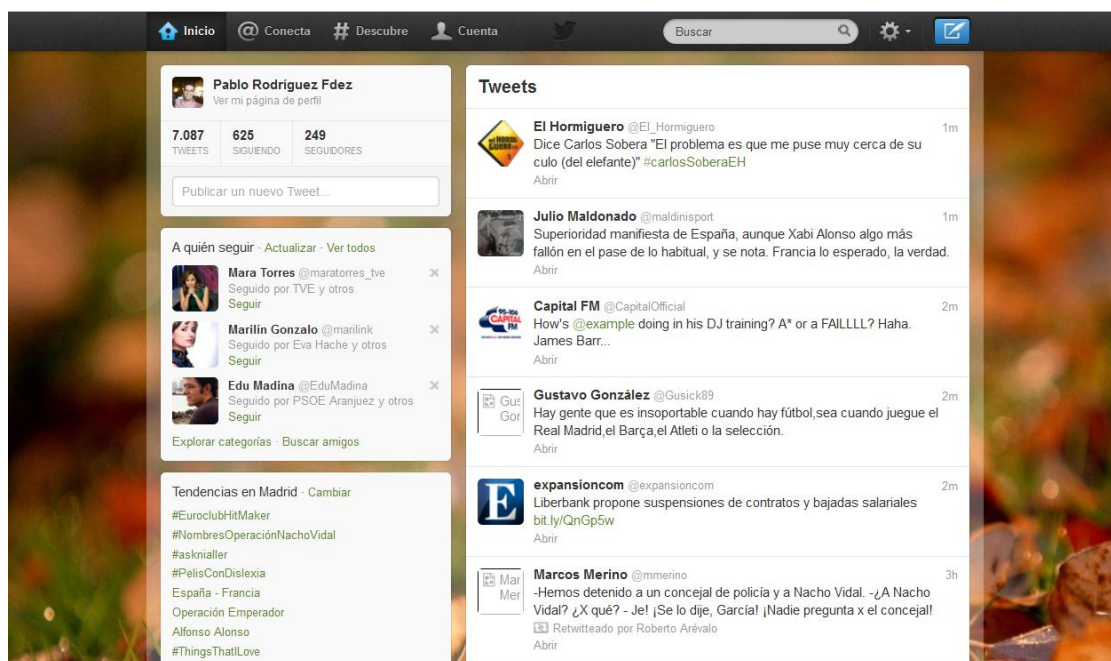


Figura 3 - La web de Twitter

2.2.2 Twitter GET API.

Esta API devuelve tweets que coinciden con una consulta específica. En concreto, devolverá datos de la cuenta con la que se ha autenticado inicialmente, ya que gestionará los datos de dicha cuenta. Se podrá extraer información de la gente a la que sigues, pero solo los 1000 primeros resultados estarán disponibles.

2.2.3 Twitter Streaming API.

Esta API permite acceder en tiempo real a varios subconjuntos de datos de Twitter, tanto públicos como protegidos, desde terceras aplicaciones.

Lo que proporciona son los estados de todos los usuarios públicos, filtrado de diversas maneras (por ID del usuario, por palabra clave, por muestreo aleatorio, según ubicación geográfica, etc.).

Para hacerse una idea, esta API es utilizada por TweetDeck¹⁴ y Echofon¹⁵, entre otras, y muestra al instante todas las actualizaciones que se realizan en la red de forma que no hace falta volver a recargar la búsqueda.

Los resultados que devuelve, son un superconjunto del resultado de la Search API que se comentaba anteriormente. A diferencia de éste, la Streaming API no realiza ningún filtrado por relevancia o por clasificación. Todos los estados que pasan el filtro de resultados de calidad están disponibles en Streaming API.

Al igual que la Search API, el formato que devuelve es JSON¹⁶ (JavaScript Object Notation), puesto que es mucho más compacto y el análisis se simplifica en gran medida por el parámetro delimitado.

A continuación se citan algunas de las formas de acceder a la información mediante las APIs. Todas operan de la misma forma pero cada una de ellas con una propuesta diferente.

¹⁴ www.tweetdeck.com

¹⁵ www.echofon.com

¹⁶ www.json.org

2.2.3.1 User Streams.

Con esta API se puede actualizar en tiempo real los datos necesarios de la aplicación. De esta forma, que podremos tener los datos de la búsqueda que se han realizado, actualizados en tiempo real, obteniendo los ‘tweets’ que los usuarios van publicando en el instante.

No tendrán límites a la hora de publicar los nuevos datos de tal forma, que el administrador de la aplicación no tendrá que administrar la frecuencia y el uso de los nuevos datos que le llegan.

Para su funcionamiento requiere que el usuario esté autenticado en la aplicación. Tiene como limitación que solo puede hacer una conexión por cuenta en dicha aplicación al mismo momento, no se puede tener varias cuentas autenticadas y recibir todos los datos en una misma conexión.

2.2.3.2 Site Streams.

Está orientada para fomentar las actualizaciones en tiempo real a servicios de sitios Web o aplicaciones móviles. Los eventos o actualizaciones, pueden ser transmitidos a todos aquellos que se hayan autenticado previamente en la aplicación. También proporciona información sin límites, como las últimas publicaciones, últimas menciones, últimos usuarios, mensajes directos, seguidores, favoritos, tweets, re-tweets, cambios del perfil y una lista de cambios.

Tiene ciertas limitaciones, puesto que actualmente está en fase BETA de pruebas, y pueden variar en función que se desarrolla una fase definitiva, es por ello que no se van a mencionar.

Definitivamente, esta API lo que devuelve son datos que puedan ser verificados como públicos, el resto requieren una autenticación para poder ser vistos.

2.2.4 Twitter Search API.

Esta es una API específica para realizar búsquedas de tweets en tiempo real a partir de una palabra clave indicada, y tiene varios aspectos que caben destacar:

- No es un índice completo de todos los tweets, pero muestra los tweets más recientes del momento.
- No mostrará tweets más antiguos de una semana.

CAPÍTULO 2

- Las consultas no podrán ser muy complejas, sino el buscador nos devolverá un error.
- Para buscar, no necesitamos estar autenticados con ninguna cuenta de usuario.

Las búsquedas con esta API son sencillas, tan solo hay que lanzar en el navegador la siguiente URL: `http://search.twitter.com/search.json?q=` incluyendo a continuación del signo igual la palabra clave que se desea buscar. Los resultados que devolverá serán en formato JSON. De esta forma, el desarrollador tendrá que tratar dicha información.

La API permite también diversos operadores, con los que se puede afinar aún más la búsqueda. Estos son algunos de ellos:

<i>twitter búsqueda</i>	<i>Contiene "twitter" y "búsqueda" (operador por defecto)</i>
<i>"happy hour"</i>	<i>Contiene exactamente la frase "happy hour"</i>
<i>amor u odio</i>	<i>Contiene tanto "amor" o "odio" (o ambos)</i>
<i>beer -root</i>	<i>Contiene "beer" pero no "root"</i>
<i>#haiku</i>	<i>Contiene el hashtag "haiku"</i>
<i>from:twitterapi</i>	<i>Enviado por el usuario @twitterapi</i>
<i>to:twitterapi</i>	<i>Enviado al usuario @twitterapi</i>
<i>place:opentable:2</i>	<i>Sobre el "place" en OpenTable con ID 2</i>
<i>place:247f43d441defc03</i>	<i>Sobre el "place" con ID 247f43d441defc03 en twitter</i>
<i>@twitterapi</i>	<i>Mención a @twitterapi</i>
<i>superhero since:2011-05-09</i>	<i>Contiene "superhero" y enviado a fecha "2011-05-09"</i>
<i>twitterapi until:2011-05-09</i>	<i>Contiene "twitterapi" y enviado desde el "2011-05-09".</i>
<i>movie -scary :)</i>	<i>Contiene "movie", pero no "scary", y con actitud positiva.</i>
<i>flight :(</i>	<i>Contiene "flight" y con actitud negativa.</i>
<i>traffic ?</i>	<i>Contiene "traffic" y haciendo una pregunta.</i>
<i>hilarious filter:links</i>	<i>Contiene "hilarious" y una dirección de URL.</i>
<i>news source:tweet_button</i>	<i>Contiene "news" y enviado a través del botón Tweet.</i>

Tabla 1 - Operadores de la función Search API

Excepciones que hay en algunos de los operadores:

- Para "since" y "until" no podemos usar el operador de negación "-".
- Hay que usar un formato de fecha completo "yyyy-mm-dd".
- Se asume que para las horas será desde/a 00:00 horario UTC.
- No podemos usar fechas futuras para los filtros.
- No podemos usar en palabras clave el espacio para separarlas.

Los resultados obtenidos pueden ser de dos tipos. O bien recientes, o los más populares en toda la red.

Un ejemplo de lo que devolvería esta API, si buscamos la palabra clave "twitterapi" y se solicita no más de un resultado, sería lo siguiente:

```
{  
  "completed_in": 0.012,  
  "max_id": 136536013832069120,
```

```

"max_id_str": "136536013832069120",
"next_page": "?page=2&max_id=136536013832069120&q=twitterapi&rpp=1",
"page": 1,
"query": "twitterapi",
"refresh_url": "?since_id=136536013832069120&q=twitterapi",
"results": [
  {
    "created_at": "Tue, 15 Nov 2011 20:08:17 +0000",
    "from_user": "fakekurrik",
    "from_user_id": 370773112,
    "from_user_id_str": "370773112",
    "from_user_name": "fakekurrik",
    "geo": null,
    "id": 136536013832069120,
    "id_str": "136536013832069120",
    "iso_language_code": "en",
    "metadata": {
      "result_type": "recent"
    },
    "profile_image_url":
"http://a1.twimg.com/profile_images/1540298033/phatkicks_normal.jpg",
    "source": "&lt;a
href=&quot;http://twitter.com/&quot;&gt;web&lt;/a&gt;",
    "text": "@twitterapi, keep on keeping it real",
    "to_user": "twitterapi",
    "to_user_id": 6253282,
    "to_user_id_str": "6253282",
    "to_user_name": "Twitter API"
  }
],
"results_per_page": 1,
"since_id": 0,
"since_id_str": "0"
}

```

Tabla 2 - Código Json que nos devuelve Twitter en sus búsquedas

Como se mencionó anteriormente, esta respuesta tiene un formato JSON, el cual la aplicación deberá interpretar y dar un formato para que sea legible hacia el usuario.

2.2.5 Diferencias entre Search API y Streaming API.

La principal diferencia entre ambas APIs, es que en la Search API es mucho más fácil la búsqueda que en Streaming API. En la Search API cada solicitud nos devolverá máximo 10 tweets, si no especificamos un número máximo.

Mientras que para Streaming API, se utiliza mediante una conexión segura a los servidores de Twitter, siendo necesaria una autenticación previa. No tendrá límite de tweets, por lo que continuamente se estará recibiendo nuevas publicaciones de la palabra clave que hayamos insertado.

Una ventaja con la que cuenta la Search API frente a Streaming API, es que no se puede buscar datos del pasado. Es decir, si le insertamos una fecha concreta y queremos que nos devuelva ciertos tweets, comprendidos en esas fechas, tendríamos que recurrir a Search API.

CAPÍTULO 2

Para las búsquedas, no hay grandes diferencias, salvo que para Search API, permite buscar una frase exacta con varias palabras en un orden específico, mientras que para Streaming API las palabras podrán coincidir en cualquier orden.

2.2.6 ¿Qué API vamos a usar en nuestra aplicación?

En este caso se usó la Search API, puesto que la manera de tratar la información es mucho más sencilla y no necesitamos que nos muestre datos actualizados en tiempo real, ya que va interesar analizar el contenido de la información que nos devuelve para que posteriormente, se pueda tratar con la técnica de curación de datos.

Otra razón, es que para el uso de Streaming API es necesario realizar una autenticación con una cuenta de usuario, y probablemente, para el uso y la realización de las publicaciones con esta aplicación, no se necesite realizarla.

2.3 Tecnologías

2.3.1 HTML¹⁷

Este lenguaje, cuyas siglas significan HyperText Markup Language (lenguaje de marcado de hipertexto), es usado para estructurar y mostrar el contenido en forma de texto y complementarlo con objetos como imágenes, vídeos y diversas aplicaciones existentes en la actualidad.

Es un descendiente de SGML¹⁸(Standard Generalized Markup Language o "Estándar de Lenguaje de Marcado Generalizado"), que es usado para referirse al contenido cifrado de tipo MIME¹⁹(Multipurpose Internet Mail Extensions o MIME (en español "extensiones multipropósito de correo de internet"), que son unas especificaciones dirigidas al intercambio de todo tipo de archivos (audio, vídeo, texto, etc.) en Internet.

Se escribe mediante etiquetas, indicando el principio y el fin de cada una, con el contenido entre ambas. También puede describir una apariencia, incluir scripts (mediante lenguaje JavaScript, que posteriormente describiremos) y que podrá aportar interacción a la página Web.

La primera publicación abierta para Internet de HTML, fue en 1991, publicado por Tim Berners-Lee. Contenía 22 elementos principales del HTML que describían el diseño inicial y relativamente simple de HTML, de los cuales solo quedan 13.

En 1993, Dave Raggett sacó un boceto competidor al modelo de Tim Berners-Lee, que sugería estandarizar características ya implementadas como tablas.

¹⁷ <http://www.w3.org/html/>

¹⁸ <http://www.w3.org/MarkUp/SGML/>

¹⁹ <http://www.iana.org/assignments/media-types/index.html>

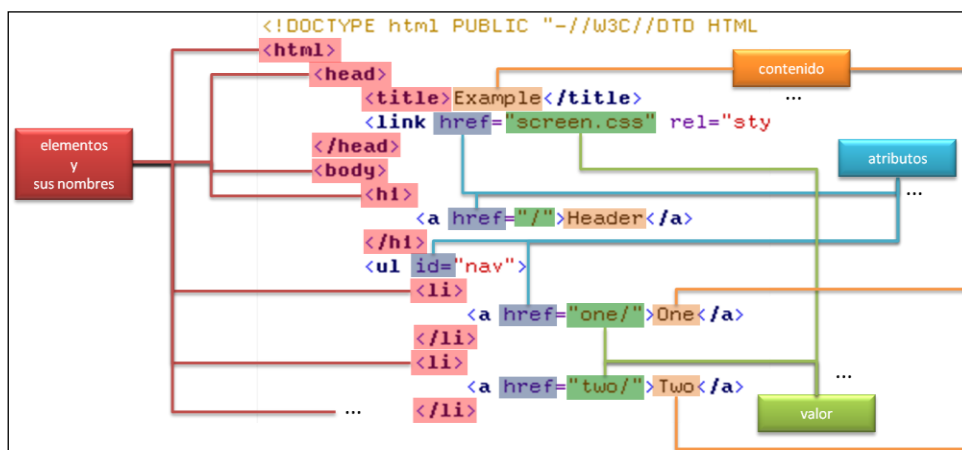


Figura 4 - Ejemplo de código en HTML5 y sus nuevas etiquetas

1. Estructura de HTML

HTML consta de varios componentes vitales que constituyen la estructura básica del sitio Web, pero también incluye elementos y sus atributos, tipos de datos y la declaración del tipo de documento.

Los elementos son la estructura básica de HTML, los cuales cuentan con dos propiedades básicas: atributos y contenido.

Todo documento HTML comienza con la etiqueta `<html>` que define el inicio del documento, para que el navegador lo interprete. Aunque no es quizás la etiqueta `<html>` la que realmente lo defina, ya que en teoría lo define el DOCTYPE:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Strict//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Tabla 3 - Cabecera de HTML5

También tendremos etiquetas del tipo `<script>` que como su nombre indica, permite incluir scripts en el sitio web. `<head>` nos define la cabecera del documento, donde está la información del documento HTML que no se muestra en el navegador. Dentro de ésta, se albergan también algunas otras etiquetas como: `<title>` el título de la página, `<link>` para referenciar hojas de estilos, funciones javascript, `<style>` para insertar el código de una hoja de estilos directamente en el documento, `<meta>` metadatos para información en una auditoría o la licencia. Y el cuerpo del HTML, que está definido por `<body>` que es la que muestra al usuario toda la información en modo gráfico a través del navegador.

Al igual que la etiqueta `<head>`, el cuerpo del documento también puede contener sus propias etiquetas interiores. Entre ellas, las más destacadas son: `<h1>` a `<h6>` para títulos y cabeceras del documento, `<table>` que son las creadoras de las tablas, pero que posteriormente exige usar `<td>` y `<tr>` para definir las celdas y filas respectivamente de la tabla. Para enlaces de hipervínculos se usa `<a>` donde se incluye un atributo con un valor `href=""index.html"` para definir la ruta dentro o fuera del sitio web. También están las etiquetas `<div>` para almacenar contenido, conocidas como división de página, y más recomendable que el uso de tablas, acompañándola de estilos de CSS.

Para crear listas tenemos ``, `` y ``, para las imágenes `` donde hay que añadirle atributos con su correspondiente valor para especificar donde está referenciada la imagen que deseamos incluir.

Y finalmente para formatos básicos en las fuentes de texto como el uso de negrita, subrayado, cursiva y tachado con ``, `<u>`, `<i>` y `<s>` respectivamente.

Esta es la versión más estándar que existe de HTML, pero posteriormente se han ido desarrollando actualizaciones del lenguaje, donde se puede interactuar y potenciar mucho más los contenidos de los sitios webs, uno de ellos se explica a continuación.



2.3.2 HTML5

La quinta versión de este lenguaje de marcado donde se crean algunas variantes a la sintaxis. El HTML clásico que sigue predominando en el código y una parte variante del XHTML²⁰ conocida como sintaxis XHTML5²¹, que deberá ser servida como XML. El lenguaje está en modo experimental, pero aun así ya está siendo soportado por la mayoría de navegadores y usado por multitud de desarrolladores web debido a los avances, mejoras y ventajas que permite.

²⁰ <http://www.w3.org/TR/xhtml1/>

²¹ <http://dev.w3.org/html5/spec/>

2.3.2.1 Nuevos elementos

Con la nueva versión de HTML, se han introducido una serie de cambios y avances en el código, creando nuevos elementos y atributos que reflejan el uso típico de los sitios webs modernos.

Muchos de ellos realizan la misma operación que `<div>` y ``, solo que tienen un significado semántico para cada función. Por ejemplo `<nav>` o `<footer>` para el bloque de navegación o el pie de página, respectivamente. Otros elementos realizan funciones a través de etiquetas estándar como `<audio>` y `<video>`.

Algunas de las mejoras introducidas en esta versión es el elemento `<canvas>`, que ha sido capaz de “*renderizar*” en algunos navegadores los elementos 3D que anteriormente no podían ser visualizados.

Otros elementos, han terminado por eliminarse, como `<center>` ``, puesto que con las hojas de estilo de CSS pueden manejarse e inclusive mejorar su funcionamiento.

2.3.2.2 Cambios en las etiquetas

Etiqueta	Atributos	Comentarios
<code><!-- --></code>	Estándar o ninguno	
<code><!DOCTYPE></code>	Estándar o ninguno	
<code><a></code>	href target rel hreflang media type	Atributo Añadido: <i>media</i> Atributo cambiado: Target
<code><abbr></code>	Estándar o ninguno	
<code><acronym></code>		Etiqueta Eliminada
<code><address></code>	Estándar o ninguno	
<code><applet></code>		Etiqueta

eliminada		
<area>	Estándar o ninguno	
<article>	Atributos globales	Nueva etiqueta
<aside>	Atributos globales	Nueva etiqueta
<audio>	autobuffer autoplay controls loop src	Nueva etiqueta
	Atributos globales	Etiqueta cambiada
<base>	Estándar o ninguno	
<basefont>	Etiqueta eliminada	
<bb>	Estándar o ninguno	
<bdo>	Estándar o ninguno	
<big>	Etiqueta eliminada	
<blockquote>	Estándar o ninguno	
<body>	Estándar o ninguno	
 	Estándar o ninguno	
<button>	Estándar o ninguno	
<canvas>	height width	Nueva etiqueta
<caption>	Estándar o ninguno	
<center>	Etiqueta eliminada	
<cite>	Atributos globales	Etiqueta cambiada
<code>	Estándar o ninguno	
<col>	Estándar o ninguno	
<colgroup>	Estándar o ninguno	
<command>	checked default disabled hidden icon label radiogroup type	Nueva etiqueta
<datagrid>	Estándar o ninguno	
<datalist>	Atributos globales	Nueva etiqueta
<dd>	Estándar o ninguno	
	Estándar o ninguno	
<details>	open	Nueva etiqueta
<dialog>	Atributos globales	Nueva etiqueta
<dir>	Etiqueta eliminada	
<div>	Estándar o ninguno	

<dfn>	Estándar o ninguno	
<dl>	Estándar o ninguno	
<dt>	Estándar o ninguno	
	Estándar o ninguno	
<embed>	height src type width	Nueva etiqueta
<fieldset>	Estándar o ninguno	
<figure>	Atributos globales	Nueva etiqueta
		Etiqueta eliminada
<footer>	Atributos globales	Nueva etiqueta
<form>	Estándar o ninguno	
<frame>		Etiqueta eliminada
<frameset>		Etiqueta eliminada
<h1> ... <h6>	Estándar o ninguno	
<head>	Estándar o ninguno	
<header>	Atributos globales	Nueva etiqueta
<hgroup>	Atributos globales	Nueva etiqueta
<hr>	Ninguno	Etiqueta cambiada
<html>	Estándar o ninguno	
<i>	Ninguno	Etiqueta cambiada
<iframe>	Estándar o ninguno	
	Estándar o ninguno	
<input>	accept alt auto-complete autofocus checked disabled form formaction formenctype formmethod formnovalidate formtarget height list max maxlength min multiple name pattern1 placeholder readonly required size src step type value width	Etiqueta cambiada: Añadidos 13 elementos a type
<ins>	Estándar o ninguno	
<isindex>		Etiqueta eliminada
<kbd>	Estándar o ninguno	
<label>	Estándar o ninguno	
<legend>	Estándar o ninguno	

	Estándar o ninguno	
<link>	Estándar o ninguno	
<mark>	Atributos globales	Nueva etiqueta
<map>	Estándar o ninguno	
<menu>	Estándar o ninguno	
<meta>	Estándar o ninguno	
<meter>	high low max min optimum value	Nueva etiqueta
<nav>	Atributos globales	Nueva etiqueta
<noframes>		Etiqueta eliminada
<noscript>	Estándar o ninguno	
<object>	Estándar o ninguno	
	Estándar o ninguno	
<optgroup>	Estándar o ninguno	
<option>	Estándar o ninguno	
<output>	form	Nueva etiqueta
<p>	Estándar o ninguno	
<param>	Estándar o ninguno	
<pre>	Estándar o ninguno	
<progress>	max value	Nueva etiqueta
<q>		
<ruby>	cite	Nueva etiqueta
<rp>	Atributos globales	Nueva etiqueta
<rt>	Atributos globales	Nueva etiqueta
<s>		Etiqueta eliminada
<samp>	Estándar o ninguno	
<script>	Estándar o ninguno	
<section>	cite	Nueva etiqueta
<select>	Estándar o ninguno	
<small>	Atributos globales	Etiqueta Cambiada
<source>	media src type	Nueva etiqueta
	Estándar o ninguno	
<strike>		Etiqueta eliminada
	Estándar o ninguno	
<style>	Estándar o ninguno	

<sub>	Estándar o ninguno	
<sup>	Estándar o ninguno	
<table>	Estándar o ninguno	
<tbody>	Estándar o ninguno	
<td>	Estándar o ninguno	
<textarea>	Estándar o ninguno	
<tfoot>	Estándar o ninguno	
<th>	Estándar o ninguno	
<thead>	Estándar o ninguno	
<time>	datetime pubdate	Nueva etiqueta
<title>	Estándar o ninguno	
<tr>	Estándar o ninguno	
<tt>	Etiqueta eliminada	
<u>	Etiqueta eliminada	
	Estándar o ninguno	
<var>	Estándar o ninguno	
<video>	src poster autobuffer autoplay loop controls width height	Nueva etiqueta
<xmp>	Etiqueta eliminada	

Tabla 4 - Las etiquetas de HTML5 y sus cambios

En esta tabla anterior, se puede ver todos los cambios que han surgido a raíz de esta nueva versión de HTML, en la que las etiquetas en amarillo serán las nuevas añadidas, en azul las que han sufrido cambios parciales o totales y en gris las eliminadas.

2.3.2.3 Novedades introducidas

Entre las novedades del lenguaje, hay que destacar las etiquetas y funcionalidades con las APIs que podemos encontrar como:

1. La incorporación de etiquetas con códecs para contenidos multimedia, como puede ser <canvas> entre otras.
2. El manejo de un gran conjunto de datos (datagrid, details, menú y command) que permiten generar tablas dinámicas que pueden filtrar, ordenar y ocultar contenido en cliente.

3. Mejoras en los formularios, como la inserción de nuevos tipos de datos como email, number, url, datetime, etc. También se presentan facilidades para validar el contenido sin utilizar Javascript.
4. Visores para MathML y SVG (fórmulas matemáticas y gráficos vectoriales). En general se deja abierto a poder interpretar otros lenguajes XML.
5. Drag&Drop que es una funcionalidad que nos permite arrastrar objetos como imágenes, tablas, contenidos... (esta funcionalidad la explicaremos, puesto que nos será de mucha utilidad).
6. Nuevas etiquetas para manejar la web semántica como header, footer, article, nav, time, link.
7. Las nuevas etiquetas permiten describir cual es su significado del contenido, la importancia, su finalidad y las relaciones que existen. No tienen especial impacto en la visualización, se orientan únicamente a los buscadores.
8. Los buscadores pueden indexar e interpretar esta “metainformación” para no buscar simplemente apariciones de palabras en el texto de la página.

2.3.2.4 APIs y Javascript²²

Aquí mencionamos algunas de las APIs más interesantes que trae esta nueva versión de HTML. Un ejemplo de ello se mencionaba anteriormente, el Drag&Drop, pero hay más novedades que también se describen a continuación:

1. Drag&Drop sirve para mover elementos mediante el ratón de una ventana a otra. Esta función, la detallamos en un apartado distinto, puesto que haremos uso de ella en la aplicación que vamos a desarrollar.
2. Podremos trabajar Offline si no disponemos en un momento concreto de conexión a internet, donde se permite descargar todos los contenidos necesarios para que trabaje sobre local.
3. Geoposicionamiento, una API muy utilizada a día de hoy por ejemplo en twitter, que para ello es necesario, que el dispositivo con el que nos conectamos pueda soportarlo.
4. “API Storage”, que también se usa para el almacenamiento persistente local, con almacenamiento de objetos por aplicación o por dominio Web (Local Storage).

²² <https://developer.mozilla.org/es/docs/JavaScript>

CAPÍTULO 2

5. Websockets es otra API de comunicación bidireccional entre páginas, algo similar a los Sockets de C.
6. WebWorkers que son hilos de ejecución en paralelo.

A continuación, mostramos un esquema, de lo que ha supuesto los cambios de HTML5 respecto de sus versiones anteriores, en lo que a estructura de etiquetas se refiere.

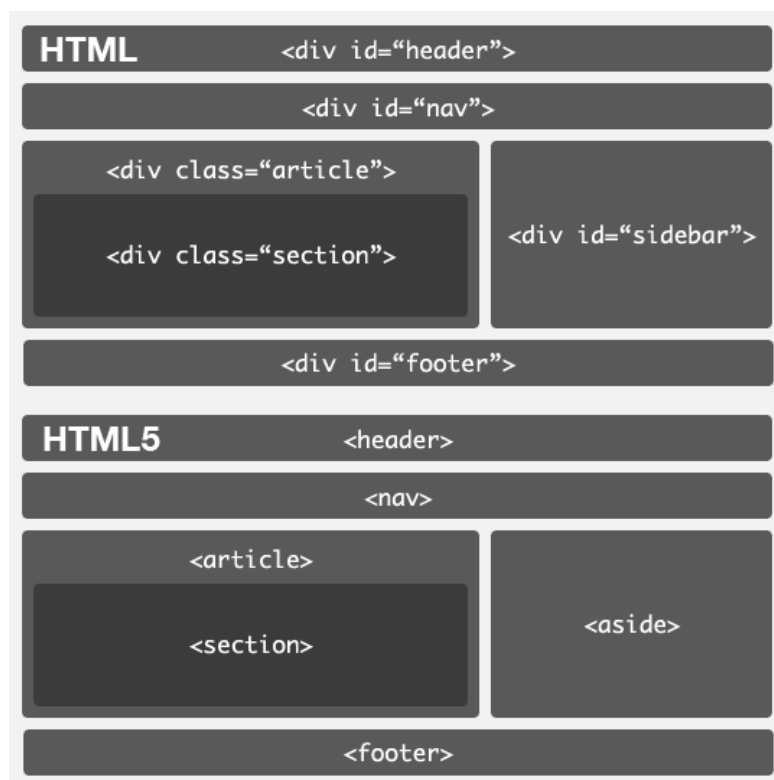


Figura 5 - Cambios en la estructura de HTML con HTML5

2.3.3 CSS²³ (Cascading Styles Sheet/Hojas de estilo en cascada)

Este lenguaje de estilos define la presentación de documentos HTML. Abarca cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo, posicionamiento avanzado entre otros. Es posible usar HTML, o incluso abusar del mismo, para añadir estilo a los sitios web desde todos los navegadores, aunque las

²³ <http://www.w3schools.com/css/>

```
body {
  margin: 4px;
  border: 3px dotted #
  font-family: sans-serif;
  color: #000000;
  background-color: #FFFFFF;
}

h1 {
  padding: 5px;
  margin: 10px;
  border: 1px solid #C0C0C0;
  color: #FF0000;
```

Figura 6 - Ejemplo de código CSS

últimas versiones como CSS3²⁴, no es compatible aún con todos ellos.

El W3C²⁵ es el organismo encargado de formular la especificación de las hojas de estilo que sirven de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento con su presentación. De este modo, la información de estilo puede ser adjuntada como un documento separado o en el mismo documento HTML en la parte de la cabecera con sus correspondientes etiquetas, o dentro de las propias etiquetas HTML indicándolo como “style” y sus correspondientes atributos y valores.

CSS se lanzó por primera vez en diciembre de 1996 con la versión CSS1, y abandonada en abril de 2008. Algunas de las propiedades que ofreció fueron las más esenciales y por las que más se le conoce, como fuente, tipo, tamaño, textos, fondos, bordes, espacios, imágenes, tablas, etc.

En su posterior versión, CSS2 fue desarrollada ya por el consorcio W3C y publicada como recomendación en mayo de 1998 y abandonada en 2008 con pocas más funcionalidades, como propiedades en las capas <div>, el concepto de “media types”, texto bidireccional, sombras, etc. Fue una ampliación de CSS1 que se terminó mejorando con la versión CSS2.1, hasta ahora la más conocida puesto que corregía algunos errores de la CSS1.

A día de hoy está comenzando a extenderse la última versión, CSS3, que como comentábamos, no es compatible por el momento con todos los navegadores. A diferencia de CSS2, esta versión, está dividida en varios documentos llamados “módulos”. Estos módulos añaden nuevas funcionalidades a las definidas en CSS2 de manera que se preservan las anteriores para mantener la compatibilidad. Algunos de estos módulos, están considerados como razonablemente estables, y sus implementaciones en los diferentes navegadores son señalados con prefijos del motor del mismo.

2.3.3.1 Sintaxis y uso de CSS

Una hoja de estilos consiste en diversas reglas que toman uno o varios selectores y un bloque de estilos con los diferentes estilos a aplicar para las etiquetas del documento que

²⁴ <http://www.w3schools.com/css3/>

²⁵ <http://www.w3.org/>

CAPÍTULO 2

cumplan con el selector que les precede. Los bloques están entre llaves donde tienen albergados una o más declaraciones de estilo con el formato “**propiedad:valor;**”.

Los dos tipos de llamadas a los selectores más comunes son las “id” y las “class”, el primero para un elemento HTML en concreto y el segundo puede abarcar varios. Los “selectors” en este caso marcan qué elementos se ven afectados por cada bloque de estilo que les siga.

```
selector [, selector2, ...] [:pseudo-class] {  
  propiedad: valor;  
  [propiedad2: valor2;  
  ...]  
}  
/* comentarios */
```

Table 5 - Sintaxis de CSS

Además del ejemplo que vemos en el cuadro anterior, se puede dar formato con el código CSS de tres formas distintas.

Una de estas formas es un **estilo en línea**, que se inserta dentro de la etiqueta HTML, aunque esta forma no es del todo correcta, puesto que alarga mucho el código de la programación web y queda poco elegante. Sí que se utiliza en casos como correos electrónicos, donde no se soportan hojas de estilo externas y es necesario hacerlo “**inline**”.

Otra forma es en una **hoja de estilo interna**, que es una hoja que está incrustada dentro de un documento HTML al comienzo del documento en la cabecera del mismo <head>, donde anuncia su comienzo con la etiqueta <style>. El inconveniente que conlleva esta forma, es que es difícil de ejecutar si se desea guardar copias sincronizadas.

Una **hoja de estilo externa**, está en un fichero independiente al HTML donde se va aplicar. Es mucho más potente porque separa las reglas de formato para la página HTML de la estructura básica de la página. Para ello, hay que insertar en la cabecera de la página (<head>) un enlace a dicho fichero, especificando que tipo de enlace estamos referenciando.

Ejemplo: <link rel="stylesheet" type="text/css" href="style/style.css" />

2.3.4 Javascript y JQuery²⁶

Aunque ambos son Javascript, podríamos decir que existen diferencias donde JQuery es una biblioteca de Javascript que permite simplificar la manera de interactuar con los documentos HTML manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

En primer lugar, Javascript es un lenguaje de programación orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se suele implementar en el lado cliente para mejoras en la interfaz del usuario y páginas web dinámicas, en bases de datos locales al navegador, aunque también tiene una versión para el lado servidor (Server-side de Javascript). A día de hoy los navegadores modernos interpretan el código JavaScript y se suele integrar en las páginas web, para interactuar con una página web.

Por otro lado JQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio. Se caracteriza además de poder seleccionar elementos DOM. Interactúa y modifica el árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath. Puede manipular la hoja de estilos CSS, realizar efectos y animaciones (personalizadas inclusive), AJAX, soporta extensiones entre otras no tan relevantes.

2.3.5 PHP (Hypertext Preprocessor)

PHP ²⁷es un lenguaje de scripting que puede ser embebido dentro de las páginas HTML. Gran parte de su sintaxis fue tomada de C, Java y Perl con un par de características específicas propias de PHP. Su objetivo principal era permitir a los desarrolladores crear páginas dinámicas con rapidez.

PHP significa Hypertext preprocessor, que aunque sus siglas confundan es así debido a que su acrónimo es recursivo.

²⁶ <http://jquery.com/>

²⁷ <http://www.php.net/>

CAPÍTULO 2

PHP/FI 2.0 es una de las primeras versiones ya fuera de desarrollo. PHP3 fue posteriormente su sucesor y mucho más “agradable” para el desarrollador. En la actualidad, la última versión es PHP5 que ofrece una amplia variedad de nuevas características.

PHP5 fue diseñado con la intención de ser compatible como las versiones anteriores, pero con algunos cambios significativos, como:

- Un nuevo modelo OOP (orientado a objetos) basado en Zend Engine 2.0.
- Nueva extensión que mejora el soporte para MySQL.
- Soporte nativo integrado para SQLite.
- Una nueva constante de reporte de errores, para sugerencias en ejecución sobre código.
- Nuevas funciones para simplificar la escritura de código.

En nuestro caso, no vamos a hacer uso de la tecnología PHP para comunicarnos entre lado cliente y lado servidor de nuestra página web, puesto que realizaremos toda la comunicación desde la página principal con el desarrollo en Javascript y Json que se comunicarán con el lado servidor.

El lado cliente es el que está trabajando de “cara al usuario” en la página principal de la aplicación, y el lado servidor es el encargado de realizar las acciones pertinentes para que podamos hacer uso de los tweets (servidores de Twitter).

Podría haber sido una opción válida también, pero hubiese hecho la aplicación más compleja y tampoco lo requiere, puesto que buscamos lo más óptimo para el desarrollo.

En un posible uso de esta tecnología durante el desarrollo de nuestra aplicación, se podría haber usado para realizar las técnicas de búsqueda y curación de datos. Para la primera tendríamos un lado cliente y lado servidor, que sería la página “index.php” en concreto la que enviaría la información de la palabra clave para buscar al lado servidor, el cual sería el encargado de buscar y mostrarnos dicha información.

Además, para el salvado de datos de la curación de datos sería necesaria la realización de algunas funciones pertinentes que se encargasen de realizar la inserción de datos en la memoria local del navegador, así como la de borrado de las historias, tanto para la “realización en caliente”, es decir, cuando estamos haciendo la curación de datos,

tanto cuando están guardadas y queremos acceder a ellas a través de la sección “historias guardadas”.

2.3.6 Las tecnologías a usar

Finalmente para el desarrollo de la aplicación se ha elegido HTML5 en vez de PHP, puesto que este último a pesar de ser más robusto, hacía más complejo el desarrollo cuando optando por HTML5 se obtiene el mismo resultado con menos líneas de código. También se han usado hojas de estilo en CSS3.

Y para las funciones, tanto JQuery como en Javascript. Se han elegido los dos lenguajes, que a pesar de ser iguales cambiando su sintaxis, se quiere mostrar la diferencia que hay entre ambas, con la simplicidad de código que tiene JQuery frente a Javascript.

2.4 Aplicaciones existentes

2.4.1 Storify

Es el medio de data curation quizás más completo y conocido hasta el momento. Permite contar historias utilizando diferentes medios sociales como tweets, fotos y vídeos. Los usuarios pueden buscar en múltiples redes sociales desde la misma página de Storify, arrastrando en el momento que lo deseen los elementos individuales en las historias que están escribiendo. De esta forma, el usuario que redacte la historia tiene toda la permisibilidad de poder cambiar el orden de los elementos con los que se cumplimenta la noticia redactada, que ayudan al lector para dar un mejor contexto.

La interfaz divide la pantalla en dos columnas, a la izquierda presenta una variedad de servicios sociales como Twitter, Facebook, Flickr, Youtube, RSS, búsquedas de Google, SlideShare, así como URLs. El panel de la derecha, es donde se introduce el título de la historia y la descripción y las fuentes de arrastre.

Luego, el contenido del lado derecho se arrastra hacia el lado izquierdo de la pantalla para ir recreando la historia. Se pueden añadir diferentes tipos como tweets, páginas de Facebook, las direcciones URLs, vídeos, imágenes, etc.

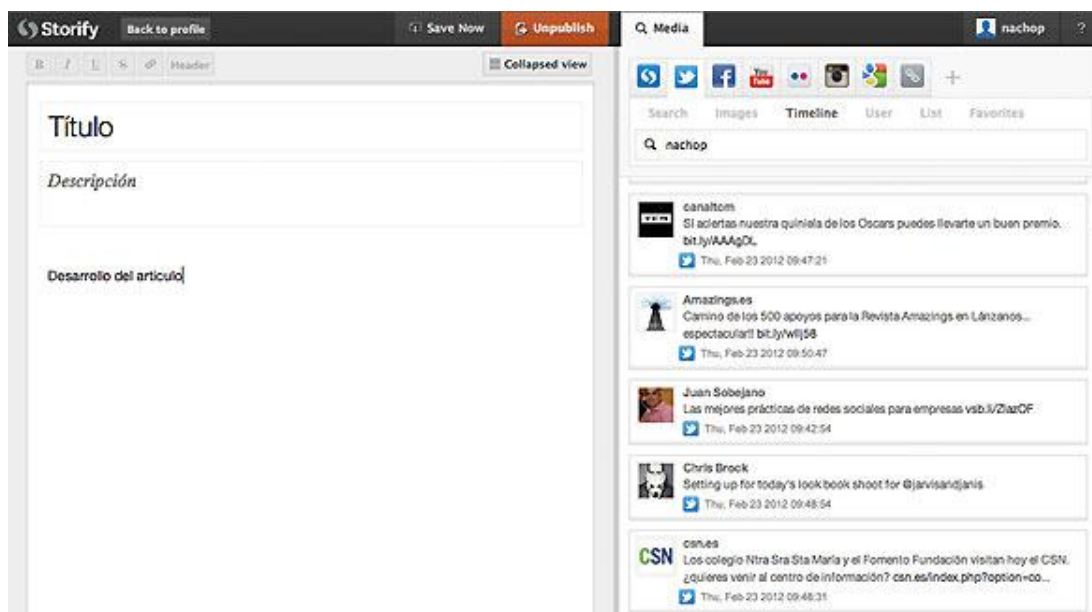


Figura 7 - Vista de la aplicación Storify

Las historias son publicadas bajo una URL que puede ser compartida para hacerla más viral o también puede ser incrustada en un blog o sitio web. Storify dispone también de estadísticas de la historia para mostrar la popularidad del contenido.

Storify también da la opción de autopublicación para las plataformas de Wordpress, Posterous, Tumblr, Twitter y Mailchimp. Cada vez que se publique una historia en Storify, automáticamente se publicará en cualquiera de estos sitios.

Una vez creada la historia, Storify, permite poderlo compartir en diversas redes sociales, como Twitter, Facebook y las mencionadas plataformas que citamos anteriormente. Además, ofrece una URL la cual podemos compartir y visitar la noticia escrita en cualquier momento. También permite realizar las modificaciones que el usuario crea convenientes en cualquier momento, pudiendo insertar, borrar tanto texto como contenido de otras redes sociales.

2.4.2 Percolate

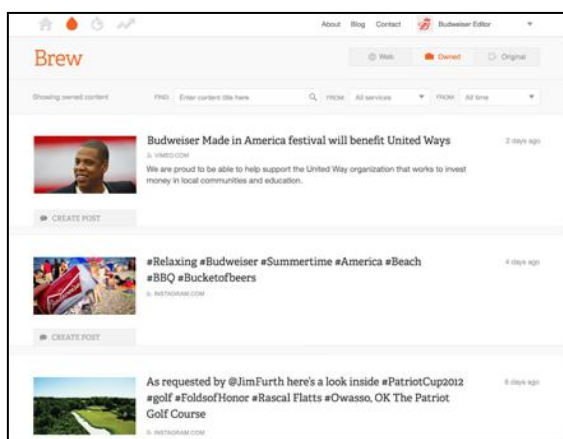


Figura 8 - Vista de la aplicación Percolate

Otra aplicación que utiliza el “data curation” es Percolate, que fue fundada en Nueva York en el año 2011.

A la hora de hacer selección de datos de otras redes sociales, podemos decir que es un poco más escueta en comparación con Storify, puesto que solo dispone de Twitter, Tumblr,

Facebook, RSS, enlaces .com, o bloques de anuncios, aunque no es el principal punto de interés en comparación a lo que se quiere hacer con esta aplicación.

Su motor de búsqueda permite hacer historias con la extracción de datos de las redes citadas anteriormente, además de hacer comentarios sobre dichas historias que el usuario crea.

Debido a las restricciones de acceso para probar esta aplicación, se desconoce si se pueden guardar datos posteriormente al hacer uso de ella.

2.4.3 Focus.io

Funciona de forma diferente a las anteriores, pero también utiliza el método de “data curation”. Para esta aplicación, es necesario registrarnos a través de nuestra cuenta de twitter. De esta forma, Focus extrae todos los seguidores que tenga cada usuario, basándose en la publicación de noticias y que estén suscritos al medio de Focus para la selección de información mediante “data curation”.



Figura 9 - Vista de la aplicación Focus.io

No nos permite por tanto, crear nuestras propias historias o noticias, sino que lo crea el portal automáticamente a través de tus intereses en Twitter. Tampoco tiene vínculos con ninguna otra red social o RSS de otras páginas.

Además puntualizar, que estos hechos son de asumir que no permitirá guardar los datos que se muestran en pantalla.

Las noticias que nos van apareciendo en la página son dos, “Summary” que será para las más relevantes y “Timeline” que van en función del momento que se han publicado.

2.4.4 Summify

Esta es otra aplicación más de esta naturaleza, que actualmente ha sido adquirida por Twitter. Debido a esta adquisición, no ha sido probable realizar pruebas de funcionamiento, pero por lo que se puede ver, realiza procesos de filtrado mediante el “data curation” en redes como Facebook, Twitter, blogs y los RSS, ofreciendo las mejores noticias/historias que se muestran en las redes ya filtradas, pero no todas las noticias/historias.



Figura 10 - Vista de la aplicación Summify

Realiza los resúmenes de noticias diarios, de forma propia, por lo que no permite al usuario crear sus propias historias. Simplemente da al usuario un resumen completo de los llamados “Trendic Topic” de Twitter o las noticias más comentadas de Facebook, etc. de las que Summify hace un escáner completo de relevancia y seguimiento por parte de los usuarios y muestra el resultado filtrado.

Tampoco permite salvar los datos de las noticias/historias creados, simplemente compartirlos, por lo que tampoco se ajusta a nuestras expectativas para la aplicación que vamos a desarrollar.

2.4.5 Comparativas

En la aplicación, llamada “Charlify”, se ha realizado algunas de las opciones de las que cuentan las mencionadas anteriormente. En una tabla comparativa, a continuación se muestra algunos de los puntos clave que se quieren para la aplicación que no todas las anteriores, lo cumplen.

Aplicación	Edita los usuarios	Buscar palabras clave	Multihistoria simultánea	Guardado de datos en LocalStorage
Storify	✓	✓	✗	✗
Percolate	✓	✓	✗	✗
Focus.io	✗	✗	✗	✗
Summify	✓	✗	✓	✗
Charlify	✓	✓	✓	✓

Tabla 6 - Comparativa de aplicaciones y nuestro proyecto

Los puntos clave en los que se va a centrar es que, sea el usuario quien pueda crear de manera propia lo que desee, buscando mediante palabras clave los tweets más deseados para completar su historia. Además también tiene la opción de poder guardar los datos (en este caso va ser en memoria Local Storage). Cuenta además, con la opción de crear más de una historia a la vez, puesto que si existe más de una tendencia en ese momento de la que estamos creando, se le da al usuario la posibilidad de crearlo en paralelo.

En cualquier caso, al que más se ajusta nuestra aplicación es a Storify, pero no llega a alcanzar los requisitos deseados por completo, como por ejemplo el guardado de datos en Local Storage. Por eso, vamos a desarrollar Charlify.

Capítulo 3

Desarrollo de la Aplicación

3.1 Introducción

En este capítulo voy a describir paso a paso las fases que se han tenido en cuenta para la implementación de esta aplicación y que menciono en el capítulo 1. Las fases a seguir son Análisis y extracción de requisitos, diseño e implementación y fase de pruebas que ya se mencionaron en el apartado 1.3.

En la fase de Análisis y extracción de requisitos se vio en el capítulo anterior (2.4) las diferentes aplicaciones ya existentes y buscamos diferencias entre éstas y la propuesta que he desarrollado con la técnica de “data curation”.

La aplicación la he llamado “Charlify”, donde el diseño de la misma y su implementación lo llevo a cabo por completo en este capítulo. En la fase de diseño, se describen las partes en las que consta la aplicación y cómo funciona el Local Storage. En la implementación se cuenta qué lenguajes se han tenido en cuenta para cada función, cómo interactúan entre ellos, la estructura que he tomado y por qué. También los diferentes ficheros que hacen que la aplicación

funcione correctamente. Además, se explica la inclusión de otras secciones como “Historias guardadas” y para qué sirven.

Y finalmente la fase de pruebas, donde se realizan algunos test a la aplicación y muestro los resultados que deben salir y como funciona.

3.2 Análisis y extracción de requisitos

En la fase de Análisis y extracción de requisitos se incluye una especificación detallada de todo lo que se ha tenido en cuenta para el desarrollo de la aplicación y sus razones. Como veíamos en el apartado 2.4, analizamos las aplicaciones ya existentes y ninguna obtenía todos los requisitos que se quiere para Charlify.

Para tener en cuenta todos los puntos importantes en los que debe consistir la aplicación se mantuvieron reuniones periódicas con el tutor del proyecto, para afinar las necesidades con las que la aplicación debía contar.

Aquí se muestra un breve resumen de los requisitos más importantes del proyecto así como algunos detalles importantes del mismo.

Requisitos funcionales:

1. La aplicación realizará todo sobre la misma interfaz

- **Detalles:** La aplicación no tendrá ningún otro fichero para realizar la búsqueda o cualquiera de las posibles opciones que se le dan al usuario, en diferentes ficheros que no sea el principal donde se realizan las búsquedas y las historias con los tweets de la palabra clave. Es decir, no tendrá un fichero cliente y otro servidor.

2. Deberá almacenarse todo en local

- **Detalles:** Las historias deberán guardarse en LocalStorage y no en ningún formato de fichero ni ningún tipo de Base de Datos, pudiendo así mismo borrar o visualizar de nuevo las historias que se han creado previamente a la selección de esta opción.

3. Utilización de tecnologías HTML5, Json, Javascript y CSS3

- **Detalles:** Para la implementación de esta aplicación solo se podrá usar las tecnologías HTML5 para la estructura web, CSS3 para los estilos y funciones como el Drag&Drop, Json para el tratamiento de los datos de Twitter y JQuery-Javascript para el guardado y borrado de datos en LocalStorage.

4. Motor de búsqueda de palabras clave

- **Detalles:** La aplicación deberá permitir la búsqueda en la red social Twitter que es a la que está ligada, con la(s) palabra(s) clave que el usuario desee mediante un motor de búsqueda que mostrará en un listado todos los tweets relacionados con esa palabra.
- **Otros:** Deberá mostrar diez tweets primeramente y si se desea mostrar otros diez más con otra palabra clave, lo harán de forma que no borre la búsqueda anterior.

5. Los usuarios crearán su propio contenido

- **Detalles:** Los usuarios podrán crear su propio contenido una vez encontrados los tweets con las palabras clave que han usado en la búsqueda, mediante la técnica de Drag&Drop de Javascript que permite el movimiento de los tweets, en este caso, hacia la zona de creación de la historia.
- **Otros:** Podrán editar también el orden así como borrar los tweets que han insertado en la historia.

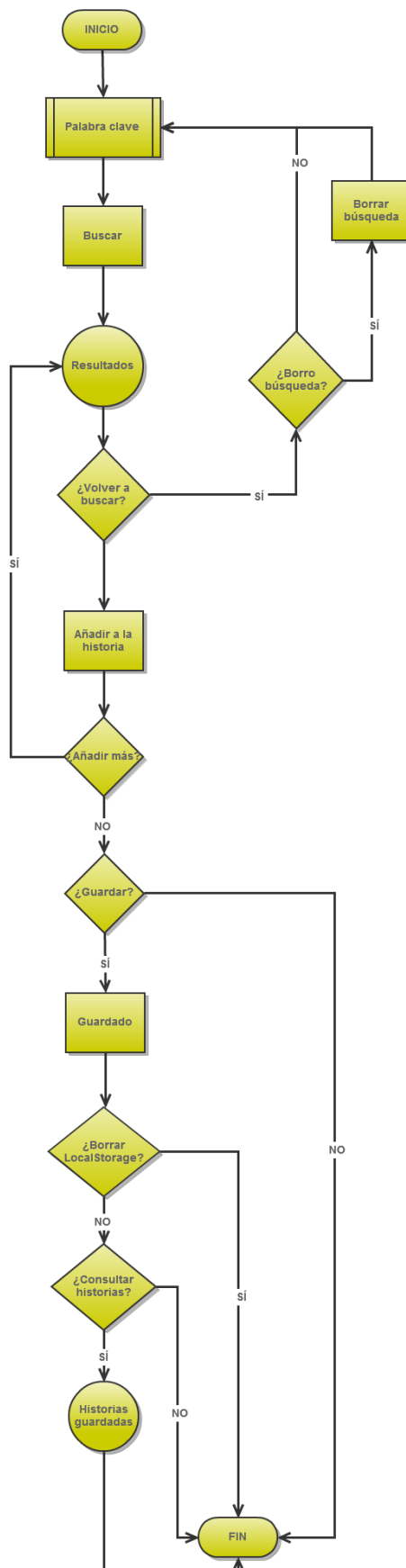
6. Creación de más una historia simultánea

- **Detalles:** Podremos crear a la vez una o más historias de la misma o diferente temática en la ventana de la página inicial. El usuario deberá seleccionar otra historia pinchando en el botón si lo desea. En caso contrario, solo aparecerá la ventana para crear una historia.

3.3 Diseño e Implementación

En el diseño e implementación se parte del análisis y la extracción de requisitos hechos previamente en el apartado anterior. Se explican los aspectos más relacionados con el diseño visual y sobre todo la lógica de funcionamiento que tiene la aplicación.

3.3.1 Diseño de las funcionalidades de la aplicación



Además del gráfico que se ve anteriormente de la funcionalidad de nuestra aplicación web, cabe destacar que es para el caso de la creación de una sola historia, pero como comento, la aplicación es capaz de hacer más de una historia como reflejamos en el diagrama de flujo que se muestra a continuación.

Este diagrama va incluido dentro de lo que es la función de Añadir a la historia, que sería la encargada del proceso de curación y que como reitero es capaz de hacer más de una.

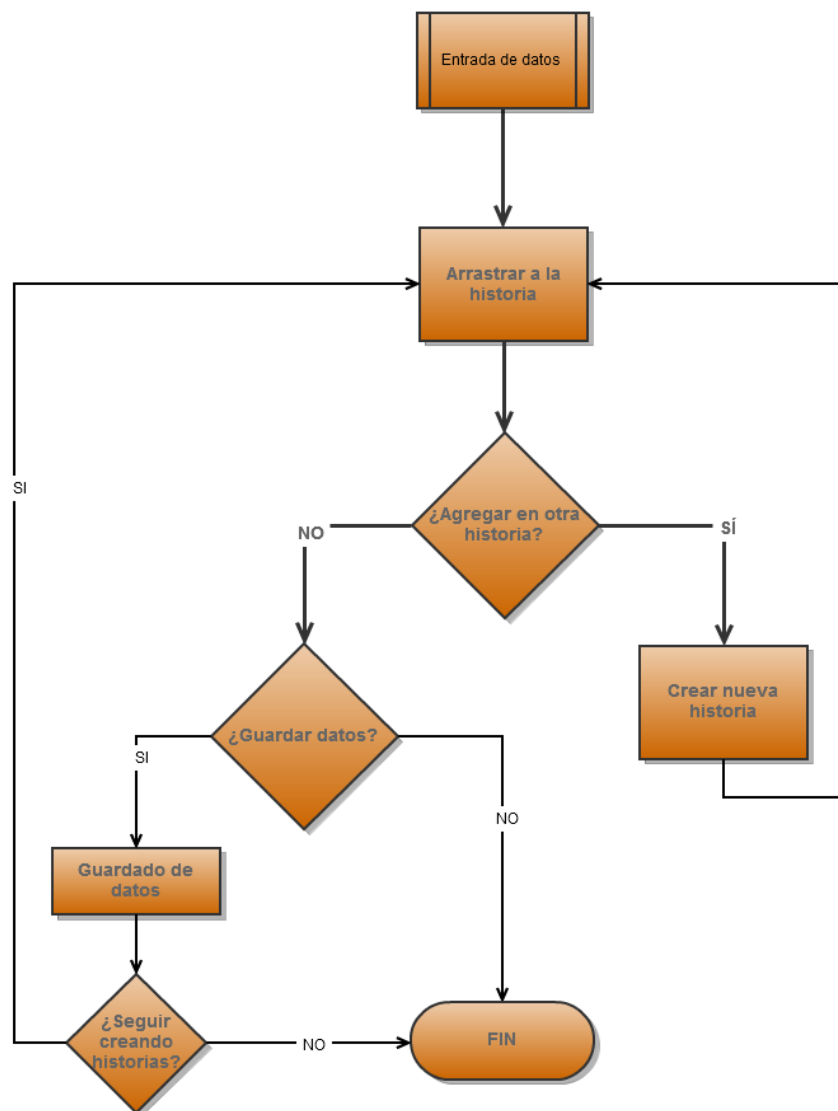


Figura 12 - Funcionalidades para la creación de una o más historias

Finalmente, una vez que ya no queramos crear más de una historia, se procedería al guardado de datos en local.

3.3.2. La interfaz gráfica de la página inicial

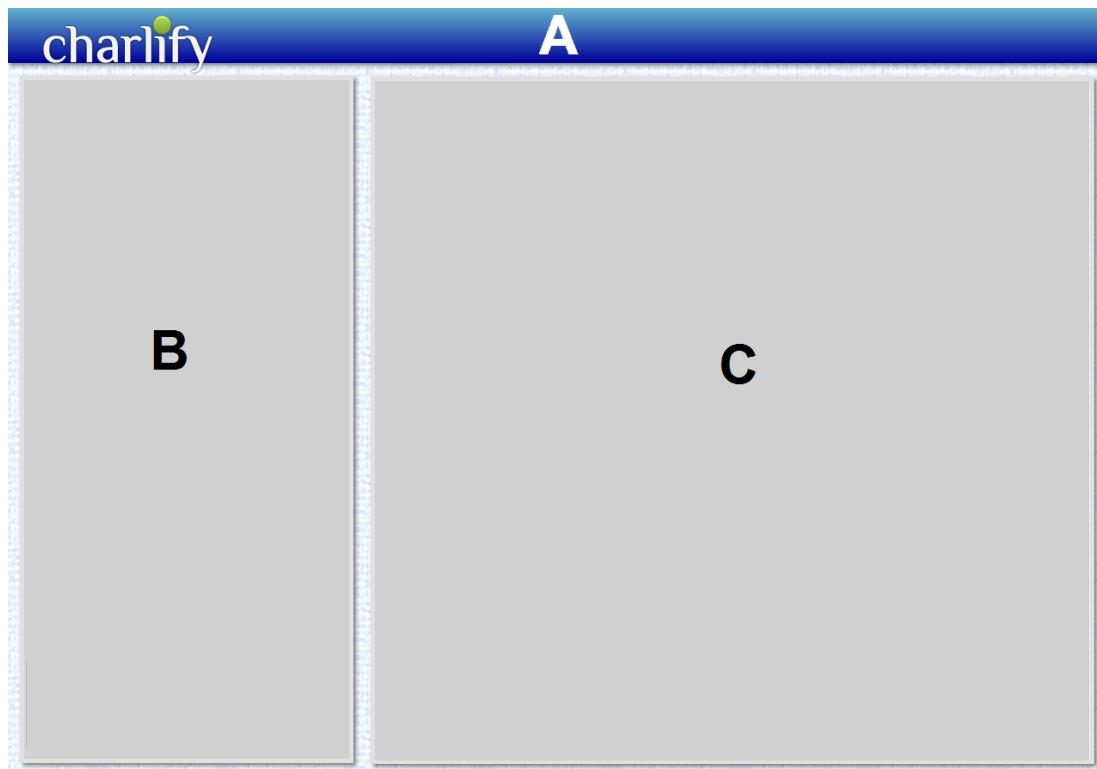


Figura 13 - Esquema o boceto de nuestra aplicación web

En cuanto a la interfaz gráfica, es la página principal que veremos y donde se desenvuelve todo el contenido y tratamiento de la información con la aplicación.

Estará distribuida en 3 importantes secciones donde podremos realizar la búsqueda, el proceso de curación y la consulta de lo almacenado en memoria local.

a) La cabecera



Figura 14 - Cabecera de la aplicación desarrollada

Imagen principal de la aplicación, donde destaca el nombre y logo (ambos el mismo), y enlaces a las diferentes secciones. Entre ellas el “Inicio”, que es donde encontramos la función principal de la aplicación y donde se realiza todo el proceso.

Otro de los enlaces que hay son las “Historias guardadas” en memoria local y por último “Acerca de”, donde incluye información al referente de la aplicación web con un breve resumen de lo que consta.

Algo que no he comentado previamente sobre la creación de la cabecera, y es que está implementado mediante listas de elementos, que son en este caso las diferentes secciones de las que consta nuestra aplicación.



Figura 15 - El motor de búsqueda

b) El motor de búsqueda

La zona de la izquierda será el motor de búsqueda de la información, donde se insertarán la(s) palabra(s) clave(s) que deseemos y posteriormente se desplegarán hacia abajo los tweets.

Como bien cito, a continuación en su código fuente, se desarrolla en JQuery que se encarga de realizar la consulta a la Search API de Twitter que devolverá un fichero de datos en Json donde se obtendrá toda la información.

En el listado de tweets que nos devuelve se pueden ver datos como la imagen del usuario, el nombre, fecha y hora de creación del tweet.

Cada tweet que se muestra, formará un “div” diferente con una “id” diferente para poder ser movido posteriormente con la función de Drag&Drop, si no

oviese una “id” distinta podría crear un conflicto con los otros “div” y no se pudiese realizar el movimiento del tweet hacia el lado de creación de la historia.

Esta búsqueda además tendrá como funciones, la posibilidad de volver a buscar nuevos tweets insertando una palabra clave nueva o borrando la existente y volviendo a realizarla.

c) La zona de curación y creación de historias



Figura 16 - Zona de curación de la aplicación

Por último la parte más importante de la aplicación, el proceso de curación haciendo drag&drop con los tweets, que se obtiene del motor de búsqueda y en el cual se seleccionan aquellos que el usuario desea para posteriormente arrastrar hacia la derecha, zona donde se crean las historias.

El proceso de arrastrado es realizado con drag&drop, función que nos proporciona Javascript y que permite incluirlo en la zona habilitada como es la parte C de nuestro esquema y que estamos explicando.

La zona habilitada para el guardado es la que se muestra en un cuadro de borde verde debajo del título de la historia y que podrá albergar tantos tweets como se quiera (con el límite del espacio disponible en Local Storage que es bastante amplio para la información que los tweets requieren).

Como se puede apreciar en la imagen, permite insertar un título que será guardado junto a todos los datos en memoria local. Para ello se ve que tenemos el botón de guardado de datos en local (“Guardar en Local Storage”) y que además también permite borrar todo el contenido de lo que tiene el navegador. ¿Por qué doy esta opción? En Local Storage como comentaba anteriormente, es una nueva funcionalidad que traen los últimos navegadores y su capacidad de almacenamiento es limitada (en función de navegador,

CAPÍTULO 3

Opera, Firefox y Chrome traen una capacidad menor al nuevo Internet Explorer 9 que es el único que lo acepta).

Además, también cabe la posibilidad de crear más de una historia en paralelo. Mientras el usuario está creando una historia, puede que encuentre algún tema relacionado que no desee entremezclar con lo que en un principio iba a realizar por el proceso de curación, por lo que se le permite crear más de una historia en un lado y posteriormente, guardarlo también como cualquier otra.

Una vez guardado los datos de la historia que se han creado, se podrá seguir creando nuevas historias restableciendo la página y si se desea también consultar las guardadas y borrarlas en la Historias Guardadas que comento a continuación.

La aplicación al completo

Finalmente, una vez definidas todas las partes y explicadas como son y que funcionalidades tienen, este es el aspecto que muestra la aplicación. Cabe destacar también la parte de Historias Guardadas que mencionaré y que supone más funcionalidades para la aplicación que he desarrollado.



Figura 17 - La aplicación al completo

Desarrollada además en su totalidad en HTML5 y CSS3 con funcionalidades en Javascript y JQuery.

3.3.2 La interfaz y funcionamiento del almacenamiento

En cuanto a la interfaz gráfica, es la página principal que se ve y donde se desenvuelve todo el contenido y tratamiento de la información con la aplicación.



Figura 18 - Página principal de la aplicación

3.3.3 La implementación

3.3.3.1 El cuerpo de la aplicación

El cuerpo de la aplicación está desarrollado con HTML5 y las nuevas etiquetas que comenté en apartados más atrás, ha servido para estructurar mucho mejor la aplicación web.

```
<!DOCTYPE HTML>
<html lang="es-ES">
  <head>
    ...
  </head>
  <body>
    <!-- Cabecera de la web -->
    <header>
      ...
    </header>
    <!-- Buscador de tweets -->
    <section>
      ...
    </section>
    <!-- Creacion de historia -->
    <aside>
      ...
    </aside>
  </body>
</html>
```

Figura 19 - Código HTML5 de la aplicación

Con la figura 19 podemos ver parte del código que dije para describir para que lo hemos usado.

La parte que está marcada como negrita, representa lo que es el código HTML básico de siempre y que se entiende que son las etiquetas necesarias para el cuerpo de una web.

En color naranja se muestra la cabecera que en nuestro caso serían los menús de la parte superior de la aplicación vista desde un navegador, se comprueba que es una de las nuevas etiquetas que se

han incluido en la nueva versión de HTML5. Entre otras están “*section*” y “*aside*” que representan el motor de búsqueda de los tweets (color verde) y la zona

donde vamos a crear una historia (color azul), que son otras dos nuevas etiquetas incluidas en HTML5.

3.3.3.2 El buscador de tweets

Esta función realizada en JQuery es la encargada de realizar la búsqueda de los tweets con las palabras clave que se han insertado. Por defecto, la API nos devolverá los últimos 10 tweets que incluyen las palabras claves introducidas.

Para la función buscar se puede ver que los datos que se solicitan es la “id” del tweet, así como la imagen del usuario, su nombre y el contenido del tweet que se devuelve según la API de Twitter en Json como se comentó anteriormente.

```
<script>
    $("#buscar").click(function () {
        var url = 'http://search.twitter.com/search.json?callback=?&q=' + $("#cadena").val() ;
        $.getJSON(url,function(json){
            $.each(json.results.reverse(),function(i,tweet){
                $("#busqueda_div").prepend('<div id="'+tweet.id+'" class="tweet"
                draggable="true" ondragstart="drag(event)"><p class="img_tweet"></p>'+<p
                class="content_tweet"><b class="title">'+tweet.from_user_name+'</b></br>'
                +tweet.text+'</br><b class="fecha">'+tweet.created_at+'</b></p></div>');
                $("#busqueda_div").slideDown("slow");
            });
        });
    });
</script>
```

Figura 20 - Código JQuery para la búsqueda de tweets en nuestra aplicación

En color marrón hemos destacado también la parte de HTML que incluye el código en JQuery para la posterior visualización correcta de los tweets tal y como se muestran en la pantalla principal de la aplicación.

Además de la función de buscar, existe otra que es para borrar el historial de tweets mostrados que limpiarán toda la ventana que muestra los tweets.

```
<script>
    $("#borrar").click(function () {
        $("#busqueda_div > div.tweet").remove();
    });
</script>
```

Figura 21 - Código JQuery para el borrado de tweets

3.3.3.3 Función Drag&Drop

Función realizada en Javascript que posibilita el arrastre de los tweets de la búsqueda al cuadro donde se desarrolla la historia que el usuario desea. En él se hace referencia a los “div” con su “id” definida que será donde se coloquen todos los que se desea.

```
<script type="text/javascript">
    function allowDrop(ev)
    {
        ev.preventDefault();
    }

    function drag(ev)
    {
        ev.dataTransfer.setData("Text",ev.target.id);
    }

    function drop(ev)
    {
        ev.preventDefault();
        var data=ev.dataTransfer.getData("Text");
        ev.target.appendChild(document.getElementById(data));
    }
</script>
```

Figura 22 - Código en Javascript para permitir el Drag&Drop de los tweets

La función “allowDrop” tendrá como objetivo habilitar el “div” donde irán los tweets que se desean arrastrar. “Drag” que serán los tweets que vamos a arrastrar y “drop” para decir donde lo vamos a dejar.

3.3.3.4 El Almacenamiento en Local Storage

Realizada también en Javascript donde primero nos comprueba si nuestro navegador es compatible con Local Storage, puesto que esta tecnología solo es compatible con los nuevos navegadores como Firefox, Chrome, Safari y las últimas versiones de Internet Explorer.

En caso de que el navegador no sea compatible, nos mostrará un mensaje que pondrá “*Local Storage Not supported in this browser*” como se muestra en la parte azul del código fuente.

```
<!--Funcion de LocalStorage-->
<script>
    var db = getLocalStorage() || alert("Local Storage Not supported in this browser");
    function getLocalStorage() {
        try {
            if(window.localStorage ) return window.localStorage;
        }
        catch (e)
        {
            return undefined;
        }
    }
    function setlocal() {

        db.setItem(document.getElementById("titulohistoria").value,
document.getElementById("twits").innerHTML += "<ul>");
        getlocal();
    }
    function ClearAll() {

        db.clear();
        getlocal();
    }
    function getlocal() {
        var res = document.getElementById("r");
        var pairs;
        var i=0;
        res.innerHTML = "";
        for (i=0; i<=db.length-1; i++)
        {
            key = db.key(i);
            res.innerHTML += "<div>"+ "Título: " + key + "</br> Tweets: "
+db.getItem(key)+"</div></br>";
        }
    }
</script>
```

Figura 23 - El código Javascript del almacenamiento en local

Con la función “setLocal” se realiza el guardado de datos de la historia que se ha creado en un último instante en la aplicación de tal modo que guardará toda la composición de datos, incluyendo así el título de la historia y los tweets.

Por otro lado, he incluido la función “getLocal” que tratará de recuperar aquello que se ha guardado y que se podrá mostrar posteriormente cuando uno quiera en la página de “Historias Guardadas” de la aplicación.

3.3.3.5 Las historias guardadas y sus funciones

Además de la vista principal de la aplicación, donde se lleva a cabo todo el proceso de curación de datos, la página de “historias guardadas” también permite hacer algunas funciones, como por ejemplo consulta, borrado parcial o borrado total de las historias que se han almacenado en Local Storage.

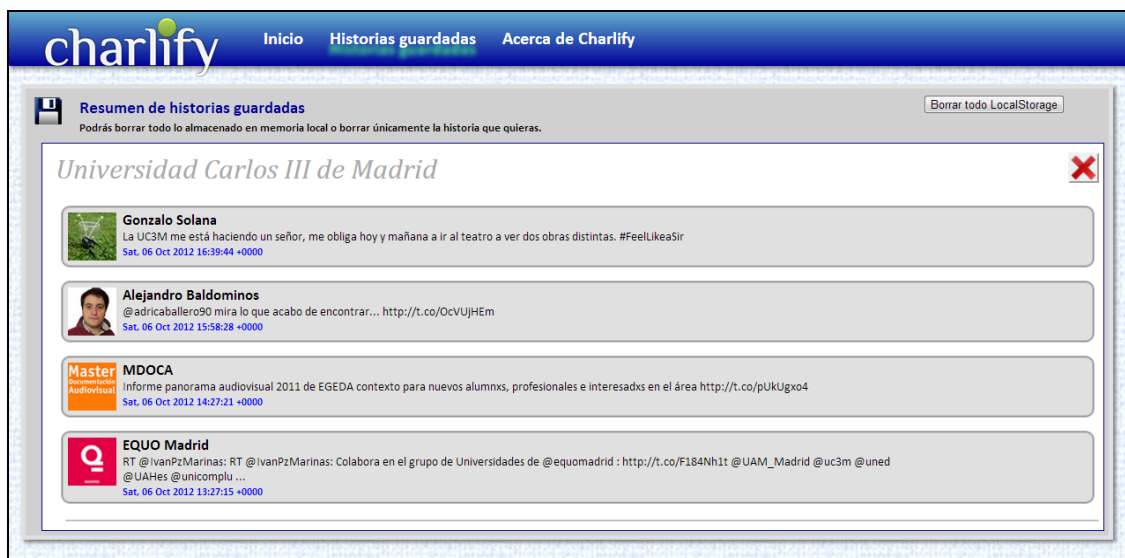


Figura 24 - Vista de las historias guardadas en la aplicación

En la figura anterior como se puede comprobar, se muestra una vista de la página con las historias guardadas. En este caso solo hay una, pero se pueden tener tantas como la memoria del almacenamiento local del navegador lo permita.

3.3.3.6 Funciones de las historias guardadas

El objetivo de esta sección no es más que mostrar como dije anteriormente las historias que se han guardado en almacenamiento local, por lo que el acceso a dichos datos es sencillo, lo comprobamos en el código que hay a continuación:

```
function getLocalStorage() {
    [...]
}

function getlocal() {
    var res = document.getElementById("div2");
    var pairs;
    var i=0;
    res.innerHTML = "";
    for (i=0; i<=db.length-1; i++) {
        key = db.key(i);
        res.innerHTML += "<div id=\"r2\" class=\"history_guardada\"> <input type=\"button\" style=\"background-image: url('images/borrar2.png'); height:35px; width:35px; float:right;\" onclick=\"deleteLS('"+key+"')\">\">\"</div>\"<p class='titulohistoria'>"+key+"</p>"+db.getItem(key)+"</hr class=\"titulohistoria\"></div>";
    }
}
```

Figura 25 - Código de la obtención de datos de Local Storage

En la figura 25 observamos como tenemos un par de funciones. La primera es para realizar una verificación de que el navegador que estamos usando tiene “LocalStorage” y que puede acceder a los datos. En segundo lugar, la función “getlocal()” es la que dará la información que se busca.

Definiendo un bucle “for” por el que se recorre todo el contenido que tiene almacenado el almacenamiento en forma de array, se irá mostrando todos los títulos y tweets de cada historia perfectamente definidos y separados por líneas, y que se podrá apreciar visualmente que la información aparece ya con un estilo mediante la inclusión de CSS3 y los elementos HTML que he introducido en el código de la función Javascript que aparece en gris y que vemos en la figura.

Además, esta sección (“Historias Guardadas”) permite también eliminar por completo todo el almacenamiento local que tiene o borrar únicamente la historia que se quiera, realizando según se muestra en las siguientes dos funciones:

```

function deleteLS(key) {
    if (confirm("\xbfEst\u00e1 seguro que desea borrar esta historia?")){
        localStorage.removeItem(key);
        location.reload(true);
    }
    else {
        return false;
    }
}

function ClearAll() {
    if (confirm("ATENCIÓN. Va a borrar todo el almacenamiento local, \xbfEst\u00e1 seguro que desea borrar todo?")){
        db.clear();
        getlocal();
    }
    else {
        return false;
    }
}
}

```

Figura 26 - Código del borrado de datos en Local Storage

En la función “deleteLS” se ve como recibe por parámetros la variable “key” que le dice cual es la historia (recibe el nombre del título) que se desea borrar en concreto.

La función “ClearAll” directamente ataca a todo el almacenamiento local, borrando por completo todo su contenido con el “db.clear”.

Además como podemos ver, antes de realizar las acciones, se ha incluido ventanas emergentes de confirmación para asegurarnos de que el usuario quiere realmente borrarlo y no ha sido un error a la hora de clicar en el botón.

3.3.4 Otras secciones de la aplicación



Figura 27 - Vista de la sección “Acerca de” de la aplicación web

Como complemento en la aplicación se ha diseñado también la sección “Acerca de”.

En ella se hace un repaso breve a toda la aplicación, como detalles de por que se realiza, los requisitos de

uso y con qué fin se crea (tanto primarios como secundarios). También se muestran una serie de capturas de pantallas, donde se explica el funcionamiento de dicha aplicación, con los pasos a seguir para un correcto funcionamiento, que se podrá ver más adelante.

Finalmente he incluido mi nombre y correo electrónico por si se quiere poner en contacto conmigo en cualquier caso.

3.4 Fase de pruebas

Durante este apartado se verá como actúa la aplicación. A continuación muestro una serie de batallas de prueba para ver que cumple todos los requisitos que se ha pedido al principio durante el análisis.

En primer lugar se requiere que nuestra aplicación fuese editada por los usuarios, donde ellos mismos pueden incluir la información que desean, podrán buscar por las palabras claves (una o varias) los tweets que desean, se podrá crear más de una historia a la vez, y algo no común y significativo en nuestra aplicación, el guardado de datos en almacenamiento local.

También el proceso de curación de datos se centra en tres pasos importantes, como son selección de contenido que atendiendo a los requisitos que cité anteriormente se cumple, edición del contenido que igual cumplimos y por último el intercambio de la información de la que disponemos, que en este caso se puede compartir mediante la visualización del contenido en la sección de “Historias guardadas” donde están todos los datos que se han creado en la historia.

3.4.1 Búsqueda de tweets

Como vemos en la figura que se muestra a continuación, para realizar la búsqueda de tweets, insertaremos la o las palabras claves que se quieren para obtener los tweets en referencia a ese tema.

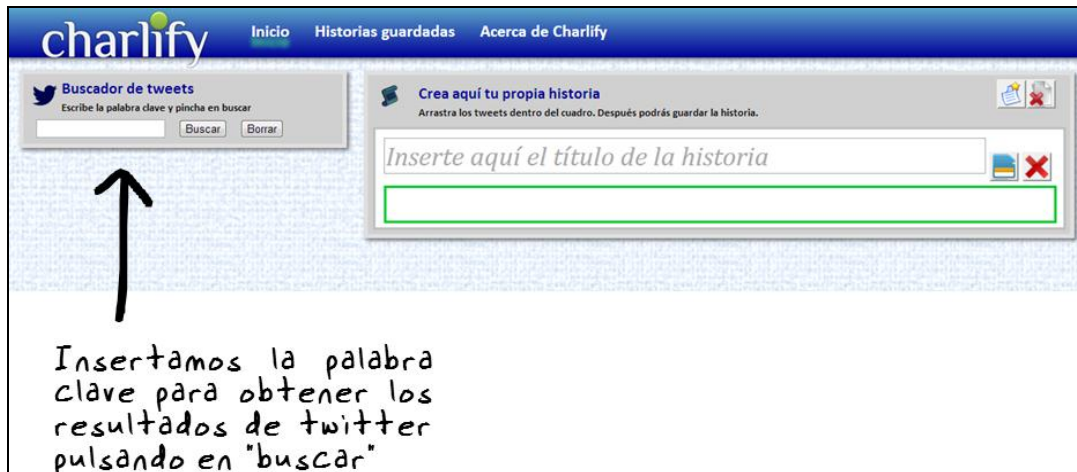


Figura 28 - Paso 1 – Búsqueda de tweets

En este caso, he tomado dos ejemplos. El primero será con una sola palabra clave y el segundo con dos.

La figura 29, que se muestra a continuación se ve como los resultados obtenidos con la búsqueda de la palabra clave “uc3m” responde a la petición de tweets que están relacionados y contienen dicha clave en el texto del tweet de cada usuario.

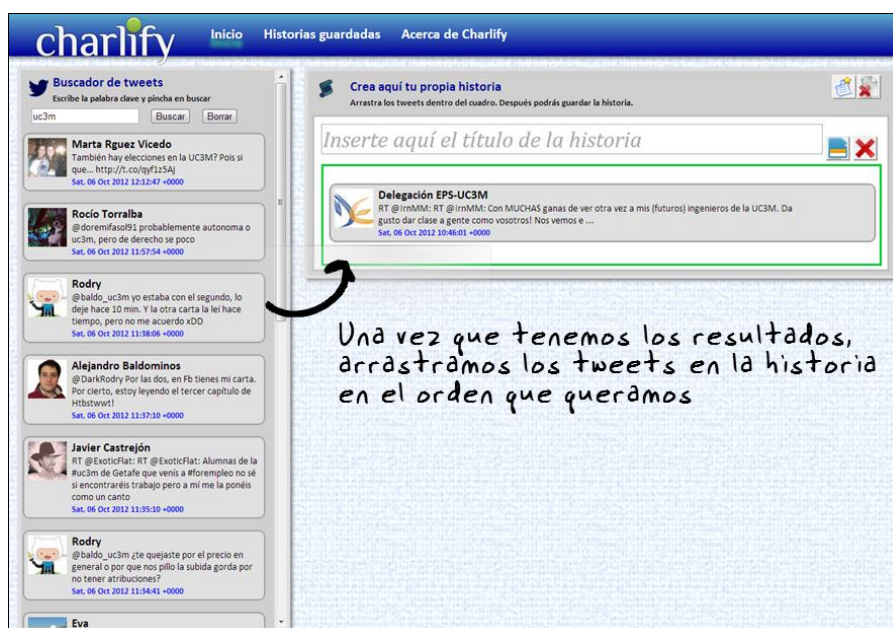


Figura 29 - Paso 2 – Resultados de una palabra clave

En el siguiente caso, vemos que he usado dos palabras clave “forempleo uc3m”, y que todos los temas que nos devuelve la función Twitter Search API son relacionados:



Figura 30 - Paso 2 – Resultados de dos palabras claves

3.4.2 Edición de los usuarios

Una vez que está la información ya buscada, procedo a arrastrar los tweets a la historia que hay a la derecha, mediante la función de Drag&Drop que he aplicado para el movimiento de los mismos, como se comenta en la figura 28. Cuando estamos realizando esta función, como se aprecia en la figura 30, el tweet quedará semi-transparente hasta que lo soltamos dentro de la historia.

Esta función de Drag&Drop satisface otra de los requisitos que cité al principio, como es la edición de las historias por parte de los usuarios según lo que deseen.

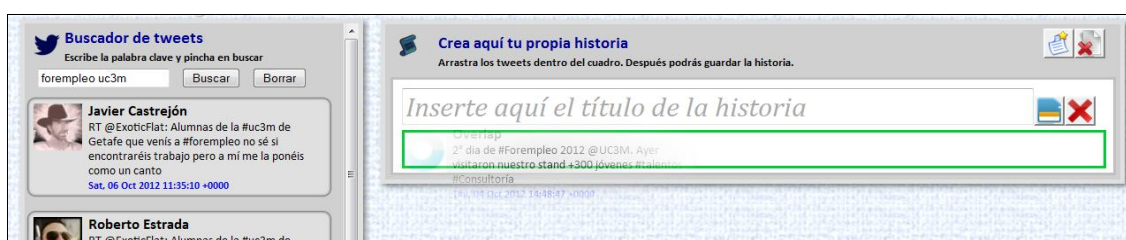


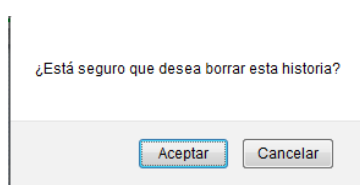
Figura 31 - Paso 3 – Arrastrado de tweets en la historia

Pero no solo el arrastrar la información que deseamos dentro de una historia es la edición que se quería para los usuarios. Algunas veces la información tan redundante puede hacer que queramos crear historias de otro tipo y que tengamos la necesidad de crearlas en paralelo y es otro de los requisitos que se tenían para la aplicación web que cumple según lo recogido en el análisis de requisitos.



Figura 32 - Paso 3 – Arrastrado de tweets con multihistorias

Otro dato que hay que tener en cuenta, es que si por un momento no nos gusta la historia que estamos creando y deseamos borrarla, tan solo con pinchar en el botón de la X roja que se sitúa en la parte superior derecha de cada cuadro de historia, se borrará la historia pero antes de esto, un mensaje emergente nos preguntará si estamos seguros que deseamos borrarlo.



Esta notificación se ha tenido en cuenta para casos en los que pulsemos el botón sin darnos cuenta y se pudiesen perder los cambios realizados o aún peor, la

historia por completo.

3.4.3 Guardado de datos

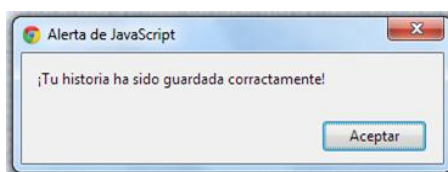
En un primer lugar hay que partir desde la figura 31, donde las historias que he creado anteriormente serán guardadas en Local Storage.



Figura 33 - Paso 4 – Guardado de datos

Como vemos en la figura que hay a la izquierda (Fig. 32), cada historia dispone de un par de botones, el de la derecha como dije anteriormente es para realizar el borrado de dicha historia y el de la izquierda que es un pequeño disquete, será para salvar la historia en almacenamiento local.

Cuando se pulsa el botón, un mensaje emergente notificará que la historia fue guardada correctamente para estar seguros que fue así y no dio error a la hora de salvarla.



3.4.4 Compartición y/o visualización de los datos

Otro punto importante en el que se centra la curación de datos, es la compartición de la información. Esta técnica se ha llevado a cabo en la aplicación de una manera significativa.

La idea desde la que se parte es que muchas de las aplicaciones existentes en la actualidad no dejan, ni compartir, ni guardar, ni visualizar posteriormente las historias o información que hemos elegido a la hora de crear la historia. Por ello, se ha elegido una

forma novedosa de hacerlo a través de “Local Storage” como he comentado en el punto anterior.

Para su posterior visualización, los usuarios de la aplicación deberán dirigirse a “historias guardadas”, una de las páginas con las que cuenta Charlify.



Figura 34 - Paso 5 – Visualización de las historias

Como se ve en la imagen, podremos visualizar todas las historias que hay. En el caso de la figura 35, tenemos dos historias pero se podrían tener muchísimas (tantas como la memoria de almacenamiento local nos permita), que posteriormente podremos manipularlas realizando la opción de borrado parcial (una a una) o por completo toda la memoria de almacenamiento local.

El borrado se realiza de forma parecida al que se realiza en el proceso de curación de los datos en la página inicial, donde si el usuario pincha sobre el botón que tiene la X en rojo en la parte derecha de cada una de las historias. Además se puede hacer un borrado completo de toda la memoria de Local Storage, clicando sobre el botón que se sitúa en la parte superior del todo en el cuadro de historias de esta sección, tal y como se muestra en la figura 35 que hay a continuación.

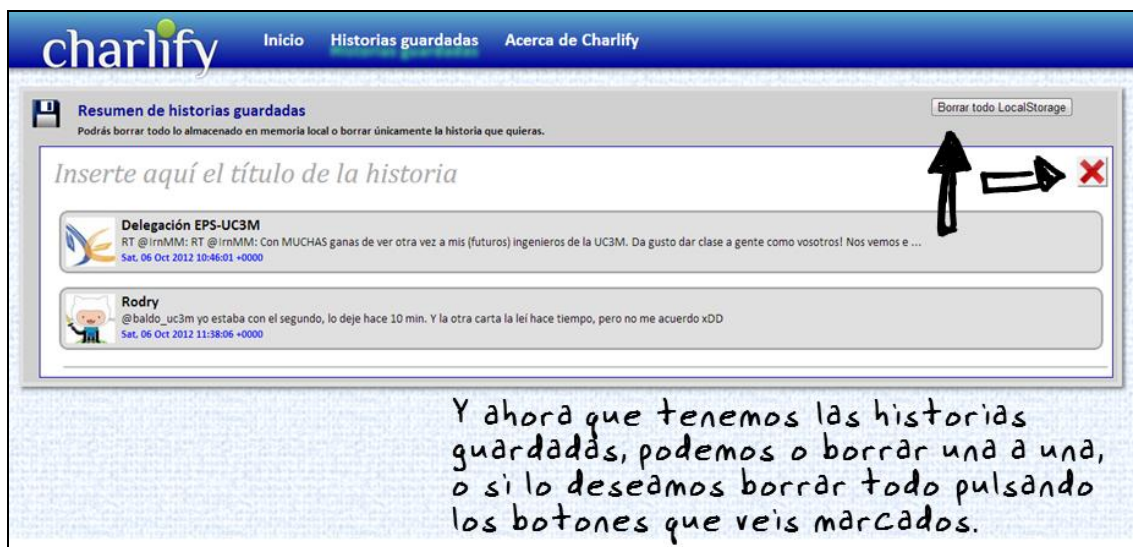


Figura 35 - Paso 5 -Visualización de las historias y botones de borrado

Y con esta última acción se cumplirían los requisitos que había para el desarrollo de la aplicación web (editar los usuarios, buscar palabras clave, multihistoria simultánea y guardado de datos en LocalStorage), así como las del “data curation” que son selección de contenido, edición del contenido e intercambio, algunas de ellas similares a las que citaba anteriormente ya que están basadas en las aplicaciones ya existentes y las funciones que dejan hacer mediante la técnica de la curación.

Con el fin de que todos los usuarios puedan hacer uso de ella y probarla, he publicado dicha aplicación con un dominio en internet en la página www.charlify.eu.

Capítulo 4

Presupuesto

4.1 Introducción

En el presupuesto vemos la parte económica del análisis, desarrollo y posproducción que hemos llevado a cabo con nuestro proyecto.

Se tendrá un desglose de los costes de personal, equipos para la realización de la aplicación y otros costes directos como han sido el hosting para el albergue en internet y el dominio de esta.

He tenido en cuenta para los costes de personal que en este caso es un Ingeniero Senior (primer año de contratación con este rango), un salario anual de 21.000 € brutos anuales, divididos en catorce pagas y que he extraído como fuente de información el buscador de empleos Infojobs.net.

Para los costes de los equipos, la información de dicha máquina ha sido extraída de la página de una tienda de electrónica en función del equipo con las determinadas características que se muestran en el presupuesto.

Y en cuanto a otros costes directos para la aplicación, se ha incluido la partida de puesta en producción de la aplicación web en internet con los costes que han sido generados para el hosting y el dominio con la empresa 1and1.es.

Finalmente en el presupuesto se ve el resumen de todas las partes que se han visto involucradas para su creación y el importe total, que se especificará con “Total Presupuesto”

4.2 El presupuesto

A continuación mostramos el presupuesto que hemos creado con los costes que acarrea el desarrollo de esta aplicación y su puesta en producción.



PRESUPUESTO DE PROYECTO

1.- Autor: Pablo Rodríguez Fernández

2.- Departamento: Informática

3.- Descripción del Proyecto:

- APLICACIÓN DE LAS NUEVAS TECNOLOGÍAS WEB PARA LA CLASIFICACIÓN COLABORATIVA DE CONTENIDOS EN REDES SOCIALES

- Duración 8 meses

Tasa de costes Indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):

14866,32 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F.	Categoría	Dedicación (hombres mes) ^{a)}	Coste hombre mes	Coste (Euro)
Pablo Rodríguez Fernández	5047262**	Ingeniero Senior	8	1.528,45 €	12.227,60 €
Total					12.227,60 €

^{a)} 1 Hombre mes = 131,25 horas. Máximo anual de dedicación de 12 hombres mes (1575 horas)
Máximo anual para PDI de la Universidad Carlos III de Madrid de 8,8 hombres mes (1.155 horas)

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Intel Core i3 2,2Ghz	585,00 €	100	8	60,00	78,00 €
Total					78,00 €

^{d)} Fórmula de cálculo de la Amortización:

$$\frac{A}{B} \times C \times D$$

A = nº de meses desde la fecha de facturación en que el equipo es utilizado

B = periodo de depreciación (60 meses)

C = coste del equipo (sin IVA)

D = % del uso que se dedica al proyecto (habitualmente 100%)

OTROS COSTES DIRECTOS DEL PROYECTO^{e)}

Descripción	Empresa	Concepto	Costes imputable
Servidor UNIX	1and1.es	Almacenamiento web	78,00 €
Dominio	1and1.es	www.charlify.eu	5,00 €
Total			83,00 €

^{e)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	12.227,60 €
Amortización	78,00 €
Costes de funcionamiento	83,00 €
Costes Indirectos	2.477,72 €
Total presupuesto	14.866,32 €

Capítulo 5

Conclusiones

5.1 Conclusiones del Proyecto

Una vez finalizada la aplicación, he llegado al punto de resolución de la problemática que existía con la sobrecarga de información de la red social Twitter y que había como objetivo primordial de reducirlo a aquello que únicamente nos interesa.

Como he descrito en gran parte de la memoria, esta solución se ha llevado a cabo gracias a la técnica de curación de datos, donde he seleccionado aquellos tweets que aportaban la información necesaria en las búsquedas que se realizan a través de la inserción de la palabra clave en el buscador que se ha implementado gracias a la Twitter Search API que la red social aporta para desarrolladores.

La posibilidad de crear más de una historia a la vez de forma dinámica (de forma que podemos crear y borrar en el momento tantas como queramos) facilita al usuario mucho más la labor de la curación de datos, puesto que en una misma búsqueda podemos encontrar varios temas que están relacionados entre sí, pero que queremos diferenciar.

Y por último el guardado de los datos en el almacenamiento local del navegador, ayuda también al usuario a que pueda recuperar dicha información cuando lo desee sin tenerlo que salvar en un fichero exportable en el PC.

Como finalidad que teníamos, era que todos los usuarios pudiesen realizar las técnicas de curación de datos ante lo que comentábamos, redundancia de información, y es por eso que hemos puesto a disposición de todos, un dominio en internet para su uso: www.charlify.eu.

5.2 Conclusiones personales

Durante esta etapa final de mi carrera desarrollando el proyecto he obtenido diversos conocimientos. Aun habiéndome adentrado en el mundo del desarrollo web, del cual ya partía con algunos conocimientos, he conseguido aprender lenguajes muy cotidianos a día de hoy en el mundo laboral, como lo están siendo el nuevo HTML5 y las hojas de estilo en CSS3. Siguiendo por lenguajes de programación para eventos y diversas funciones como Javascript y JQuery en las que no había tenido ocasión de verlas, pero que me han servido de gran valor para desarrollar mi carrera profesional y que así espero que sea, como desarrollador web.

No sin dejar de mencionar las redes sociales, que a día de hoy navego muchísimas horas al cabo de la semana y que me apasionan, y ha sido esta una gran combinación para poder unir dos fuertes de era digital que vivimos.

Como solución a la abundante información que puedo ver al cabo del día, ha sido la idea de realizar esta aplicación que me ha ayudado a comprender la importancia de la selección de información que circula a diario en ellas y en concreto Twitter, que es una de las que más auge ha conseguido en los últimos años en España.

5.3 Desarrollos y mejoras futuras

La aplicación que se ha desarrollado ha estado muy a la altura de lo que se pedía, reuniendo todos los requisitos deseados para solventar el problema del exceso de información, pero se podría haber cumplimentado con más requisitos que para un futuro sería muy positivo.

En primer lugar la búsqueda de información en la red social Twitter es de calidad ya que a día de hoy existen cientos de medios de comunicación, no solo usuarios físicos que aportan gran información, y que estos podrían cumplimentarse por ejemplo con la inserción de un texto adicional (no solo el título) que describa los hechos que estamos filtrando con la técnica de “data curation”. La aportación de información de otras redes sociales enriquecería mucho más todo los contenidos ya que algunas otras redes, como por ejemplo Youtube, tienen APIs para desarrolladores como Twitter y con restricciones de privacidad en sus contenidos al igual que pueda ocurrir con Twitter.

Por otro lado, la posibilidad de mostrar la información que se ha guardado en la memoria local es bastante buena puesto que nos permite volver a consultar aquello que llevábamos interesados desde un cierto tiempo, y que con la aplicación desarrollada hemos conseguido filtrar. En este caso, para hacerla más dinámica se podría recurrir a la posibilidad de una función de impresión o exportación de un fichero de cada historia si el usuario lo desea.

Y finalmente creo que si estos puntos que se comentan se llevan a cabo, Charlify estaría muy a la altura de las aplicaciones existentes que hay a día de hoy en la red.

Glosario

ASCII	<i>American Standard Code for Information Interchange</i>
CSS	<i>Cascading Style Sheets</i>
HTML	<i>Hypertext Markup Language</i>
IAB Spain	<i>Interactive Advertising Bureau Spain</i>
JSON	<i>JavaScript Object Notation</i>
MIME	<i>Multipurpose Internet Mail Extensions</i>
PHP	<i>Hypertext Pre-processor (acrónimo recursivo)</i>
RSS	<i>Really Simple Syndication</i>
SGML	<i>Standard Generalized Markup Language</i>
SVG	<i>Scalable Vector Graphics</i>
TT	<i>Trendic Topic</i>
URL	<i>Uniform Resource Locator</i>
W3C	<i>World Wide Web Consortium</i>
XHTML	<i>eXtensible HyperText Markup Language</i>
XML	<i>eXtensible Markup Language</i>

Referencias

[1] Data Curation:

<http://crlh.acrl.org/content/71/4/194.full>

[2] España ranking mundial redes sociales:

http://www.profesorbot.com/noticias/servicios_web/resumen-informe-ontsi-sobre-redes-sociales-en-espana/

[3] Profundizar Twitter:

<http://angeles-donos-requena.suite101.net/profundizando-en-las-redes-sociales-el-top-ten-de-twitter-a65979>

[4] Mapa Redes Sociales en España:

<http://www.socialblabla.com/wp-content/uploads/2012/01/mapa-redes-sociales.jpg>

[5] Data Curation:

<http://datosconinteligencia.blogspot.com/2010/08/que-es-la-data-curation.html>

http://en.wikipedia.org/wiki/Data_curation

[6] ¿Qué es la Curación? (recurso en francés):

<http://www.20minutes.fr/web/666430-web-la-curation-nouvelle-tarte-creme-web>

[7] Curación es arte:

<http://www.optimalaccess.com/blog/curation-art>

[8] Imagen de estructura de HTML5 y sus cambios:

<http://www.alistapart.com/articles/previewofhtml5>

[9] CSS:

<http://es.html.net/tutorials/css/lesson1.php>

[10] Javascript:

<https://developer.mozilla.org/es/docs/JavaScript>

[11] JQuery:

<http://www.jquery.com>

[12] PHP:

<http://es2.php.net/manual/es/faq.general.php>

[13] Storify:

<http://spanish.fastrack.media.com/blog/post/storify-como-crear-historias-a-partir-de-la-web-social/>

[14] Percolate:

<http://percolate.com/about/>

[15] Summify:

<http://en.wikipedia.org/wiki/Summify>

Proyecto fin de carrera de Pablo Rodríguez Fernández en la Ingeniería Técnica en
Informática de Gestión para la Universidad Carlos III de Madrid

- www.charlify.eu -

- 26 de Octubre de 2012 -



Universidad
Carlos III de Madrid